



Universitat de Lleida
Escola Politècnica Superior

Disseny d'una aplicació Android de l'EPS seguint el Model de Procés de la Enginyeria de la Usabilitat i Accessibilitat.

Treball Final de Carrera.

Enginyeria Tècnica d'Informàtica de Gestió.

Autor: Miquel Casals Pelegri

Director: Toni Granollers Saltiveri.

Lleida a Setembre de 2014.

Motivació

Aquest treball neix de la necessitat de realitzar un projecte de final de carrera i de la iniciativa personal d'iniciar-me en la creació i programació d'aplicacions Android. Es realitza amb el model de procés de l'enginyeria de la Usabilitat i Accessibilitat per realitzar la continuació del bloc optatiu de IPO i així aprofundir també en el desenvolupament d'aquesta tècnica.

Agraïments

Havent finalitzat aquest projecte, vull agrair primerament a la Bet, la meva parella, el suport que m'ha proporcionat en tot moment. Sense els seus ànims, motivació i sacrifici personal, no hagués arribat fins aquí. Donar les gràcies, també, a l'Oriol i a la Montse, que han hagut de suplir el meu lloc com a pare en molts i molts moments.

Agrair al Toni la confiança i la seva infinita paciència en tot el projecte.

Per últim, un especial agraïment als meus pares per l'esforç que han hagut de realitzar perquè jo pugues gaudir d'aquesta oportunitat d'avançar en la meva formació acadèmica i professional.

...dedicat a les dos criatures que m'han fet una mica
més complicada la finalització d'aquest projecte.

Resum

Aquest document descriu el procés de creació d'una aplicació basada en la pàgina web de l'Escola Politècnica Superior amb tecnologia Android seguint el model de Procés de l'Enginyeria de la Usabilitat i Accessibilitat, desenvolupat per Griho, des de la idea inicial fins a la programació i presentació d'aquesta.

Índex de continguts

1. OBJECTIUS	1
2. PLANIFICACIÓ I EQUIPAMENT	3
2.1. Planificació	3
2.2. Equipament.....	5
3. ESTRUCTURA DEL DOCUMENT.....	7
3.1. Introducció.....	7
3.2. Metodologia i tecnologia.	7
3.3. Anàlisi de requisits.....	7
3.4. Disseny.	7
3.5. Tests amb usuaris.....	8
3.6. Programació.....	8
3.7. Conclusions.....	8
3.8. Treballs futurs.	8
4. METODOLOGIA I TECNOLOGIA.....	9
4.1. Disseny d'interfícies per a dispositius “petits” i mòbils	9
4.1.1. Els dispositius.....	9
4.1.2. Mida i densitat.....	9
4.1.3. Punts clau en el disseny d'interfícies per a mòbils.....	13
4.2. Disseny Centrat en l'Usuari seguint el model MPlu+a.	15
4.2.1. MPlu+a.....	15
4.3. Android.....	17
4.4. Eclipse.....	19
5. ANÀLISI DE REQUISITS.....	21
5.1. Implicats.....	21
5.2. Usuaris.....	22
5.2.1. Perfils.....	22

5.2.2. Personas	23
5.3. Rols	25
5.4. Enquesta sobre els hàbits d'ús de la pàgina web de l'EPS.....	26
5.4.1. Qüestionari	26
5.4.2. Qüestionari final.....	27
5.4.3. Resultats.	27
5.4.4. Anàlisi dels resultats	27
5.5. Anàlisi contextual de tasques.....	29
5.5.1. Tasques principals.	29
5.5.2. Tasques opcionals.	29
5.5.3. Anàlisi de "la competència".	30
5.6. Plataforma.....	35
5.6.1. Selecció de Plataforma.	35
5.6.2. Versió.....	36
5.6.3. Pantalla.....	37
5.7. Objectius.....	39
5.7.1. Objectius funcionals.....	39
5.7.2. Objectius d'usabilitat	40
5.7.3. Objectius d'accessibilitat	41
6. DISSENY.	43
6.1. Arquitectura de la informació / definició de les seccions.	43
6.1.1. Card sorting.....	44
6.1.2. Estructura definitiva.	46
6.2. Prototip de paper.	47
6.3. Storyboard navegacional.	57
6.3.1. Versió sense menú.	57
6.3.2. Versió amb menú.....	58
7. TESTS AMB USUARIS.....	59
7.1. Preparació.	59
7.1.1. Material.....	59
7.1.2. Selecció de software i configuració.	60
7.1.3. Selecció de la mostra.	62
7.1.4. Plantilla del test.	63
7.1.5. Recorregut amb prototip digital amb Menú.	65
7.1.6. Recorregut amb prototip digital sense Menú.	67
7.2. Resultats.....	70

7.2.1.	Valoracions dels usuaris en el test.....	70
7.2.2.	Anàlisi dels resultats.	72
7.2.3.	Conclusions del test.	74
8.	PROGRAMACIÓ.....	75
8.1.	Gestió del projecte.	75
8.2.	Desenvolupament	76
8.2.1.	Menú inicial	76
8.2.2.	Matrícula i Guia Docent.	79
8.2.3.	Agenda i Notícies.....	80
8.2.4.	Preferències.	98
8.2.5.	Pantalla de presentació (<i>Splash</i>).	105
8.2.6.	Calendari.....	106
8.2.7.	ActionBar	107
8.3.	Iconografia.....	112
8.3.1.	Icona de l'aplicació (<i>Launcher</i>).....	113
8.3.2.	Icones del menú.....	114
8.3.3.	Icones de la barra d'acció.	114
9.	CONCLUSIONS	117
10.	TREBALLS FUTURS	117
10.1.	Calendari personalitzat.....	119
10.2.	Correu electrònic de l'EPS.	120
10.3.	Sincronització amb el Campus Virtual.	120
10.4.	Implementació en multi-plataforma.	120
10.5.	Elements d'accessibilitat pendents.	120
10.6.	Llançament de l'aplicació.....	121
11.	BIBLIOGRAFIA.....	123
12.	WEBGRAFIA.....	125
13.	ANNEX I. FORMULARI DE L'ENQUESTA.	127
14.	ANNEX II. RESULTATS DE L'ENQUESTA.	133
15.	ANNEX III. PROTOTIP DIGITAL.	147

1. Objectius

- Disseny d'una aplicació per a dispositius mòbils de l'Escola Politècnica Superior. L'accés a una pàgina web des d'un dispositiu mòbil comporta accedir a una interfície dissenyada per a monitors de 15" o més però a través d'una pantalla de 4". Aquest fet implica que la navegació dins la pàgina, lectura de text o selecció d'enllaços és bastant complicada i en alguns dispositius de gama baixa pot inclús trigar a realitzar la càrrega de tots els elements visuals. El primer objectiu d'aquest projecte és dissenyar una futura aplicació per aquests dispositius.
- Millorar la comunicació entre estudiants, professors i secretaria per adaptar-la a l'autèntica utilitat dels dispositius mòbils, que és la immediatesa. Es vol crear un canal de comunicació que en temps real es pugui, entre altres coses, saber si s'anul·la una classe, el material necessari per una sessió de laboratori o saber en uns instants on es realitza la conferència que vols assistir i no recordes l'aula.
- Màxima informació en el mínim espai. L'aplicació ha de ser personalitzada per tenir sol la informació que ens interessa ocupant el menor espai possible de memòria en el dispositiu mòbil i facilitant l'accés a les dades de forma més ràpida.
- Llançament de l'aplicació. Es pretén posar en circulació l'aplicació per tal que tots els usuaris de l'escola en puguin fer ús i simultàniament posar-la a prova per a futures modificacions. És possible que aquest punt no sigui possible en el temps marcat fins a la presentació però hi ha el compromís d'assolir-lo encara que sigui amb el projecte entregat.

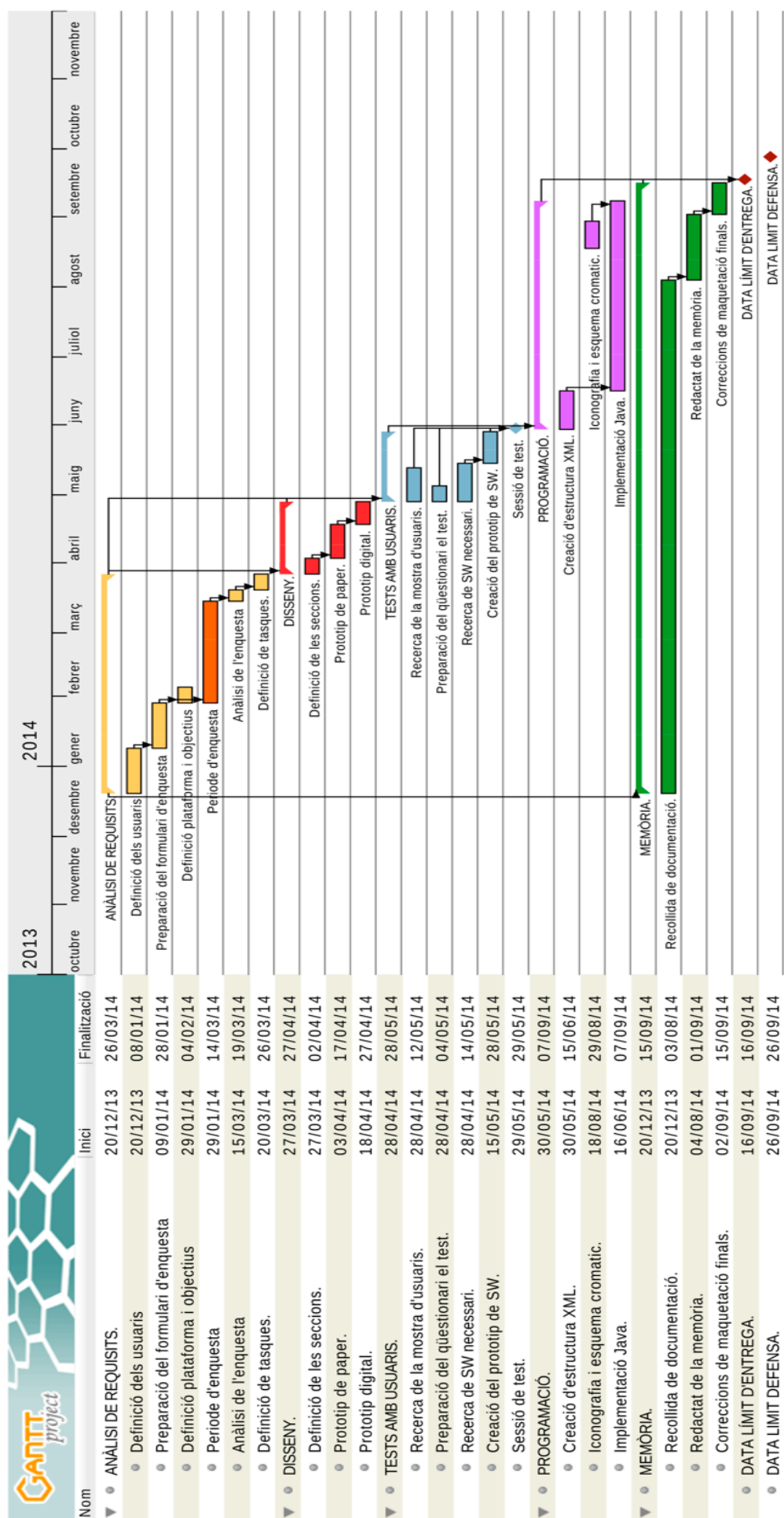
2. Planificació i equipament.

2.1. Planificació

Es preveu la possibilitat de dedicar una mitjana de 10h setmanals a la realització d'aquest projecte. La planificació es realitza amb la previsió dels dies necessaris per cada punt del projecte.

Treball	Durada	Treball	Durada
Definició usuaris	20	Qüestionari	7
Preparació de formulari	20	Recerca de SW	17
Plataforma i objectius	7	Configuració de SW	14
Període d'enquesta	45	Test amb usuaris	1
Anàlisi enquesta	5	Estructura XML	17
Definició de tasques	7	Programació Java	84
Definició Seccions	7	Iconografia i estil	12
Prototip de paper	15	Recollida de documentació	Tot el treball
Prototip digital	10	Redactat de memòria	29
Selecció de mostra	15	Correccions de memòria	14

Taula 1. Durada prevista dels diferents punts de treball.



Gràfic 1. Diagrama de Gantt.

2.2. Equipament.

Per la realització d'aquest projecte es necessita el següent equipament:

Material.

- Portàtil o PC. El disponible és Macbook Pro d'Apple. 13" de pantalla, processador Intel Core i7 2,8GHz i 4GB de RAM.
- Mòbil Android. Adquirit per l'ocasió Motorola Moto G. 4,5" de pantalla, Processador quad-core A7 1.2GHz, 1GB de RAM i 8GB de memòria interna
- Connexió a Internet. No és imprescindible si ja es disposa del SW necessari instal·lat en el portàtil però molt sovint es necessita realitzar alguna consulta i facilita molt la disponibilitat d'aquesta.

Software.

- Eclipse o qualsevol ADT (Android Developer Tools). Es podria fer la programació amb un simple editor de text i amb línia de comandes però aquest software és totalment gratuït i facilita moltíssim el desenvolupament.
- SW de disseny gràfic. Per a la realització de tota la iconografia és aconsellable disposar d'algun programa de disseny. En el nostre cas es disposa de Photoshop CS5.

3. Estructura del document

3.1. Introducció.

Part introductòria del treball. Definició dels objectius, calendari i els motius pels quals es realitza aquest projecte.

3.2. Metodologia i tecnologia.

Breu explicació de les bases del sistema operatiu Android, del Model de Procés de l'Enginyeria de la Usabilitat i Accessibilitat (MPlu+a) i del software Eclipse amb el qual s'ha desenvolupat la part de programació. S'inclou també un apartat d'explicació dels punts a tenir en compte quan es dissenya per a dispositius "petits" com telèfons.

3.3. Anàlisi de requisits.

Es defineix els usuaris, els rols i les tasques que realitzaran i s'estudia les versions d' Android i les densitats de pantalla. Es realitza una enquesta en l'àmbit de l'EPS per conèixer els hàbits d'ús dels usuaris amb el dispositiu i amb la pàgina web. Aquest apartat ens serveix per definir quines funcions tindrà l'aplicació i quina informació de la pàgina web incloem.

3.4. Disseny.

En l'apartat de disseny es mostra el procés d'interpretació dels requisits trets del punt d'anàlisi de requisits, com es dissenya la interfície gràfica i la navegació entre seccions. Es defineix com es distribueix la informació que es vol incloure a l'aplicació i finalment, es realitza el prototip de paper i la maqueta digital que després s'utilitzarà per fer el test amb els usuaris.

3.5. Tests amb usuaris.

Es realitza un test de recorregut cognitiu amb un grup d'alumnes de la EPS. En concret, crea un qüestionari, es selecciona una aplicació que ens permet realitzar el test amb el prototip digital en el mateix dispositiu mòbil, es realitza el test i s'analitza les dades obtingudes. Aquest test ens serveix per acabar de decidir la versió de maqueta que es desenvoluparà amb Android.

3.6. Programació.

Conversió de la maqueta digital a una aplicació Android. S'explica l'estructura que es decideix utilitzar per cada secció per tal que s'adapti al disseny previst. També es descriu els punts on s'han trobat dificultats i les correccions que s'han realitzat, perquè i com.

3.7. Conclusions.

Conclusions del treball.

3.8. Treballs futurs.

Indicació de les possibles millores o línies d'ampliació del projecte.

4. Metodologia i tecnologia.

4.1. Disseny d'interfícies per a dispositius "petits" i mòbils

4.1.1. Els dispositius.

Actualment en el mercat hi ha gran varietat de dispositius i cadascun amb les seves propietats. Existeixen des de rellotges de polsera amb una pantalla de 3" fins a televisors de més de 50" passant per tauletes, mòbils, dispositius de geolocalització, etc.

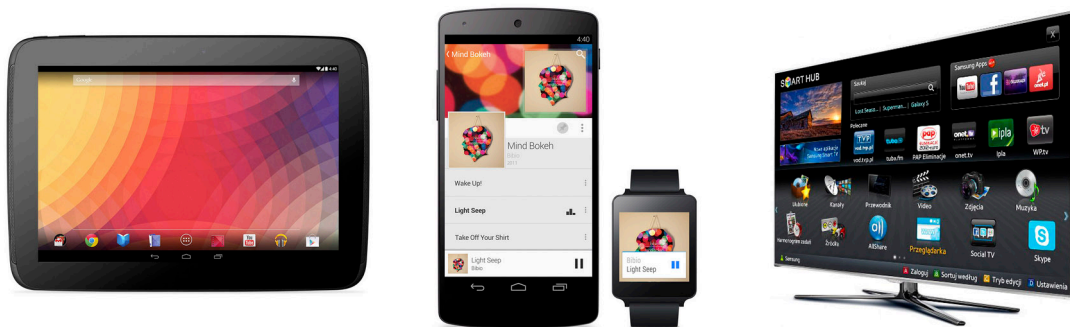


Figura 1. Dispositius que tenen S.O. Android. D'esquerra a dreta: Tauleta Nexus 10, telèfon intel·ligent Nexus 5, rellotge G-Watch i televisió Samsung.

Aquest estudi es centra en els dispositius mòbils i es deixa de banda els altres elements, tot i que alguns dels punts que es tracten també serien hi aplicables.

4.1.2. Mida i densitat.

Cada dispositiu té les seves característiques de hardware i software. Si ens centrem en les pantalles, no sol varien les dimensions sinó que també varia la densitat. La mida de la pantalla es defineix en polsades i la densitat per un nombre de píxels per polsada. Com a dissenyador s'ha d'optimitzar els recursos

per tal que cada dispositiu funcioni de la millor manera possible i es visualitzi correctament.

En el moment que s'interactua en un dispositiu tàctil es fa amb el dit que, salvant les diferències físiques entre usuaris, té unes dimensions notables si es compara amb un cursor tipus ratolí. La mida de tots els components de la interfície ha de ser suficient per poder dur a terme una acció com prémer un botó al primer toc sense problemes. Igualment la mida de les fonts tipogràfiques no pot ser massa reduïda, ja que tot i ser un dispositiu "petit" la vista humana té uns límits i s'ha de poder llegir el que es mostra a la pantalla.

En la web de *Android Developers*¹ es considera que la mida mínima per un component tocable són 7mm. Com que cada dispositiu té la seva mida i densitat de pantalla, aquests 7mm correspondran a un nombre de píxels. Per simplificar el disseny i no perdre'ns en un mar de xifres entre densitats i mides, s'utilitzen unes unitats diferents que serveixen per definir una unitat estàndard en tots els dispositius. En Android existeixen les següents unitats:

Unitat	Descripció original	Traducció	Descripció
px	Pixel	Píxel	"El pixel [...] puede definirse como la más pequeña de las unidades homogéneas en color que componen una imagen de tipo digital" ² En altres paraules, un píxel seria cada punt d'un únic color que compona la pantalla.
In	Inch	Polsada	Unitat de mesura internacional que equival a 25,4 mm. És la unitat de mesura que s'utilitza en les pantalles.
DPI	Dots Per Inch	Píxels per Polsada	És la unitat bàsica de densitat de pantalla. Segons alguns estudis la màxima nitidesa que percep l'ull humà son 300 ppp (píxels per polsada).
MDPI	Medium Density	Densitat mitjana	160 DPI. També hi ha LDPI, HDPI, XHDPI i XXHDPI. Cadascuna amb la seva proporció DPI (veure Figura 2 i Figura 3)

¹ <http://developer.android.com/design/style/metrics-grids.html>

² <http://definicion.de/pixel/>

DP / dp	Density-independent Pixel	Píxel independent de la densitat	Cada pantalla té diferents DP en funció solament de la mida però sempre els mateixos si la densitat és diferent. Per una pantalla MDPI utilitza 160 DPI per cada DP però en una XHDPI per cada DP seran 320 DPI.
SP / sp	Scale-Independent Pixel	Píxel independent de la mida.	És una unitat que s'utilitza bàsicament mida de lletra. A diferència de la unitat DP aquesta s'escala en funció de la mida de pantalla. D'aquesta manera en mòbils de diferents mida de pantalla la lletra mantindrà la mida real visible. És important programar la mida de les fonts tipogràfiques en aquesta unitat perquè s'ajusta als canvis de preferències dels usuaris.

Taula 2. Unitats utilitzades per Android per a gestió de mides dels diferents elements de la pantalla.

En la Figura 2 es pot visualitzar més gràficament la diferència que hi ha entre les densitats. Amb una mateixa mida en *dp* cada densitat té diferent quantitat de píxels per banda i diferent qualitat d'imatge.

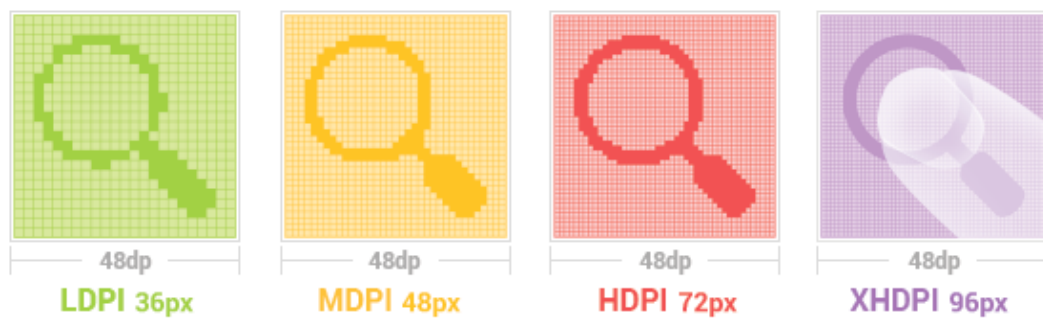


Figura 2. Diferències en la mida i la quantitat de píxels depenent de la densitat en programar amb *dp*.



Figura 3. Mida real dels elements si no es tingués en compte les densitats i tots els píxels fossin iguals.

En la Figura 3, en canvi, es pot comprovar la diferència de mida de les icones en cas que tots els píxels fossin de la mateixa mida, és a dir, en lloc de treballar amb *dp* ho féssim amb píxels (*px*).

Amb aquestes unitats, Android indica que la mesura recomanable per als elements de pantalla és *48dp* i amb els marges de *8dp* entre elements com es pot apreciar en la Figura 4 que ens mostra les mides d'un botó i una icona.

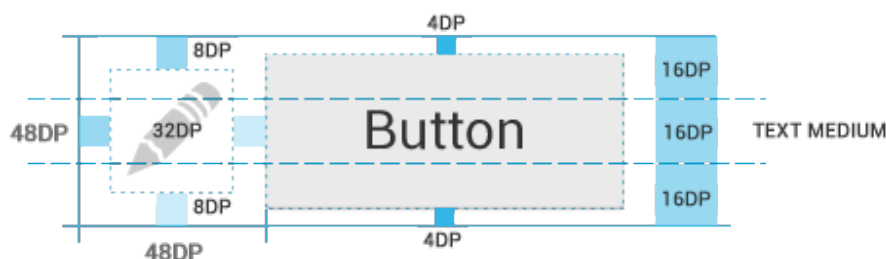


Figura 4. Esquema de mides per crear un botó amb una icona.

En la Figura 5 es pot apreciar com queda una interfície amb les mides que recomana Android.

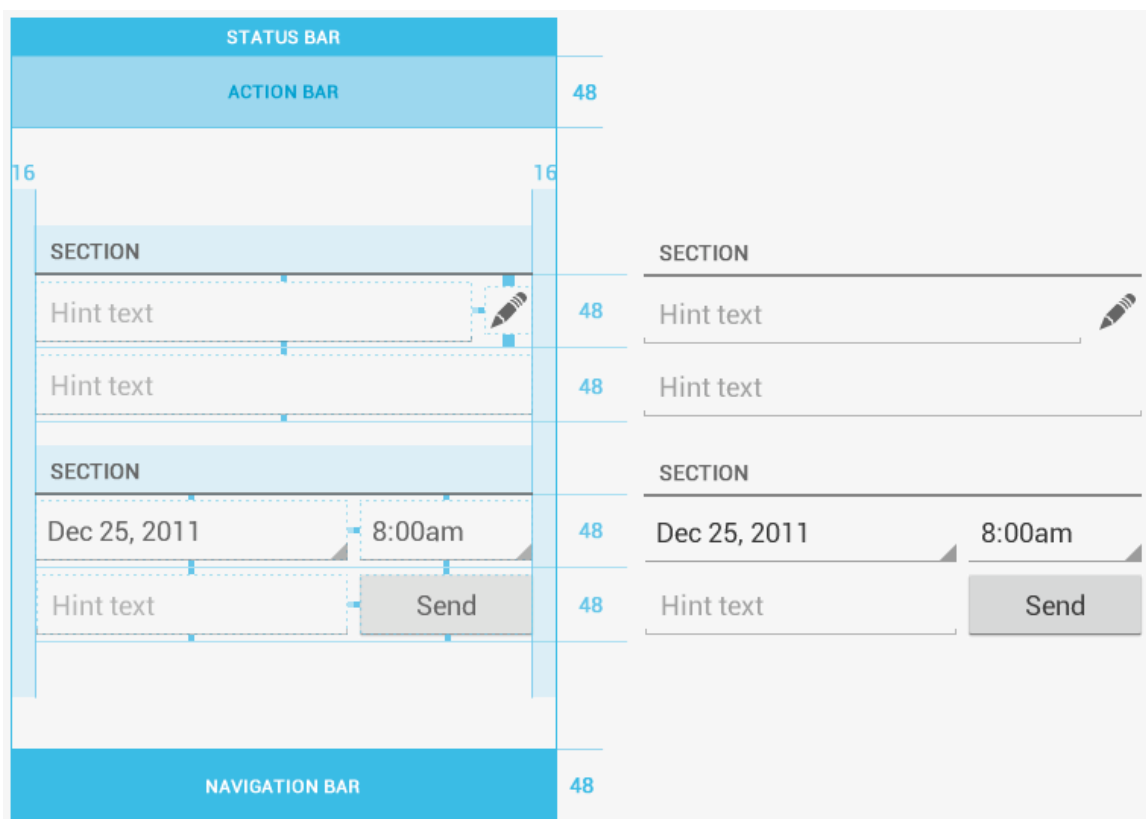


Figura 5. Exemple d'interfície amb les mesures recomanades per Android a l'esquerra i el resultat d'aquesta a la dreta.

4.1.3. Punts clau en el disseny d'interfícies per a mòbils.³

"Bona navegabilitat, igual que el bon disseny, és invisible" [3]. Amb això el que Theresa Neil vol dir és que si realitzem correctament el disseny de la interfície no és perceptible però en canvi, si realitzes una interfície mal distribuïda i difícil d'entendre, sí que ho serà però molt negativament perquè els usuaris no seran capaços de realitzar la seva tasca o els hi suposarà un esforç major. Alguns punts importants a tenir en compte serien els següents:

- **Zones tocables de mida adequada.**

A diferència de dissenyar aplicacions per a PC, tauletes o navegació web, dissenyar per a dispositius mòbils té una limitació d'espai important i juntament amb què la interacció es fa amb el dit implica que els elements han de tenir una mida mínima per poder ser útils sense ser massa grans per deixar espai als altres elements. Android recomana 48 *dp*, encara que com s'ha vist anteriorment, aquest valor podria ser lleugerament inferior.

- **Optimització de la pantalla.**

És important aprofitar bé l'espai de pantalla perquè és limitat. Una capçalera, un títol o qualsevol element ocupa un espai que hem de valorar si realment és necessari. Cal incloure únicament la informació més important i de forma clara i directa. S'ha de valorar la quantitat d'informació que introdueixes per no obligar a desplaçar massa vegades el contingut de la pantalla. Hi ha l'opció de crear un sistema de pestanyes per tal que pugui arribar a destí de forma més ràpida.

- **Navegació simple i clara.**

Cal intentar que l'usuari arribi a la informació desitjada amb el mínim de passos possibles i que hagi d'introduir el mínim de dades de forma manual. Evitar el que es diu "Navegació infinita". És un format que permet estar sempre obrint noves finestres sense fi. És important que la navegació tingui un punt de partida i un

³ Més informació a les pàgines web <http://www.nngroup.com/articles/infinite-scrolling/> i <http://www.taptapnetworks.com/blog/5-claves-de-usabilidad-en-tu-app>

final i els moviments entre seccions es facin a través d'un menú. Afavoreix que l'usuari sàpiga on es troba en tot moment.

- **Cada acció comporta un resultat.**

Sempre que l'usuari premi un botó és clau que el sistema respongui i que l'usuari sàpiga que l'ha premut encara que sigui un missatge en pantalla indicant-ho. Si no rep aquesta resposta, pot pensar que el botó no funciona i per tant l'aplicació tampoc o que no s'ha realitzat l'acció correctament i probablement ho repeteixi uns quants cops. En el mateix punt, vigilar els temps de càrrega. La paciència és molt limitada avui dia i els temps d'espera en aplicacions de dispositius mòbils són crítics.

- **La primera impressió és clau.**

Com a últim punt, cuidar molt bé la presentació de les pàgines inicials. S'ha de tenir en compte que l'usuari que utilitza la nostra aplicació per primer cop i veu que no compleix les seves expectatives l'eliminarà i n'instal·larà una altra que faci el mateix o similar. És molt important aquest punt perquè si es crea una aplicació segurament n'hi haurà d'altres que facin unes funcions similars. És molt difícil captar usuaris / clients i s'ha de procurar que no se'n vagin per una mala impressió inicial.

4.2. Disseny Centrat en l'Usuari seguint el model MPlu+a.

El Disseny Centrat en l'Usuari (DCU) és un concepte que, com diu el nom, basa el projecte en la participació de tots els usuaris en tot el procés. Habitualment l'enginyeria del software definia la realització d'una aplicació des del punt de vista dels programadors i es feia testos amb els usuaris quan ja el producte estava molt avançat. El DCU amplia aquesta participació per tal que des dels inicis, en el moment que es recopila la informació necessària i es realitzen els primers dissenys ja es faci consultant tots els usuaris. Sabent com l'usuari desenvolupa les tasques que es vol transformar en la nostra aplicació, l'entorn on les desenvolupa, les possibles dificultats o discapacitats que pot haver, etc. es pot realitzar una aplicació molt més fidel al seu model de treball. De la mateixa manera estalviarà costos i assegurarà l'èxit de l'aplicació. Si es realitzen proves amb els usuaris en el moment que l'aplicació ja està en una fase molt avançada, les modificacions que s'haurien de fer seran molt més costoses. En canvi, el cost seria molt menor si s'hagués fet els testos progressivament en cada pas del desenvolupament on els prototips són molt més barats i fàcils de modificar.

4.2.1. MPlu+a.

DCU és un concepte que no té unes directrius acceptades per tota la comunitat a l'hora d'aplicar-ho. Un model existent que marca un procés per fer-ho és el Model de Procés de l'Enginyeria de la Usabilitat i l'Accessibilitat (MPlu+a) desenvolupat per GRIHO. En la seva pròpia pàgina web defineixen l'objectiu com *"... conseguir «casar» el modelo de desarrollo de sistemas interactivos de la Ingeniería del Software con los principios básicos de la Ingeniería de la Usabilidad y los de la accesibilidad proporcionando una metodología que sea capaz de guiar a los equipos de desarrollo durante el proceso de implementación de un determinado sistema interactivo."*⁴

⁴ <http://www.grihotools.udl.cat/mpiua/fases-mpiua/>

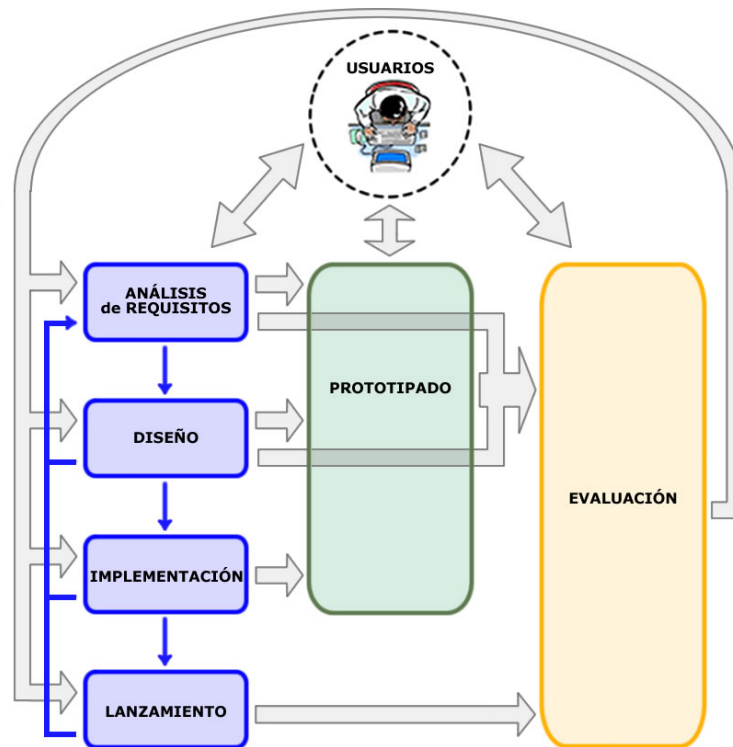


Figura 6. Esquema MPlu+a.

El model MPlu+a (Figura 6) divideix el procés en diferents fases separades en tres pilars i l'usuari com a element central i que participa en totes elles. Totes les fases estan relacionades per unes fletxes. Pot semblar que no marquen cap ruta clara entre fases però aquesta és la clau del model, la possibilitat de repetir els cops necessaris cada fase. Les fletxes blaves més primes és el procés que marca tradicionalment l'enginyeria del software i les grises més gruixudes indiquen on intervé l'usuari. D'aquest procés en diuen el Mètode iteratiu.

El primer pilar de color blau representa el procés clàssic de l'enginyeria del software. Es realitza tota la recopilació d'informació dels usuaris en l'Anàlisi de requisits, el disseny, la implementació per a la plataforma desitjada i la finalització. A diferència del procés clàssic, en aquest model s'inicia el prototipat i avaluacions amb la participació de l'usuari des de la primera fase repetint el procés les vegades que sigui necessari.

4.3. Android.

Android és un Sistema Operatiu pensat inicialment per a dispositius mòbils però que després s'ha estès a tot tipus d'aparells com ja s'ha comentat anteriorment.

El 2005 Google compra el sistema operatiu, que era pràcticament desconegut, a una petita companyia. Al novembre de 2007 es crea la Open Handset Alliance que aglutina Google i diferents fabricants de Mòbils, hardware i operadors de telecomunicacions per crear la primera versió d' Android i el paquet de desenvolupament de software (sdk). Des d'aquell moment el nombre de dispositius amb Android ha anat en augment fins que a finals de 2010 ja era el sistema operatiu més utilitzat del món en dispositius mòbils.

El que diferencia Android respecte a altres sistemes operatius com són iOS, Windows o Blackberry OS és que Android és un sistema multi-plataforma de desenvolupament lliure i de codi obert basat en Linux 2.6. Això vol dir que qualsevol pot obtenir el codi font, modificar-lo i utilitzar-lo gratuïtament on vulgui. El fet que la programació sigui en Java, tot i que també es pot fer amb codi natiu (C++), i que l'entorn de programació amb Eclipse sigui gratuït també i multi-plataforma ha permès el ràpid augment d'aplicacions disponibles en la botiga virtual de Google "Google Play Store".

L'arquitectura Android està plantejada en cinc capes (Figura 7). Les capes superiors basen el funcionament en les inferiors. Les aplicacions que es realitzen són la capa superior (Applications) i se serviran dels diferents gestors que Android proporciona a la segona capa denominada Marc d'aplicació (Application Framework). Aquests gestors al mateix temps se serveixen de les llibreries de la tercera capa realitzades en codi C (Libraries). Al mateix nivell del sistema hi ha la capa d'execució on hi ha la Maquina Virtual Dalvik i les llibreries pròpies en codi natiu (Android Runtime). En el nivell més bàsic (Linux Kernel) hi ha el kernel 2.6 de linux lleugerament modificat per ser adaptat a dispositius mòbils amb menors recursos que els ordinadors. [1]

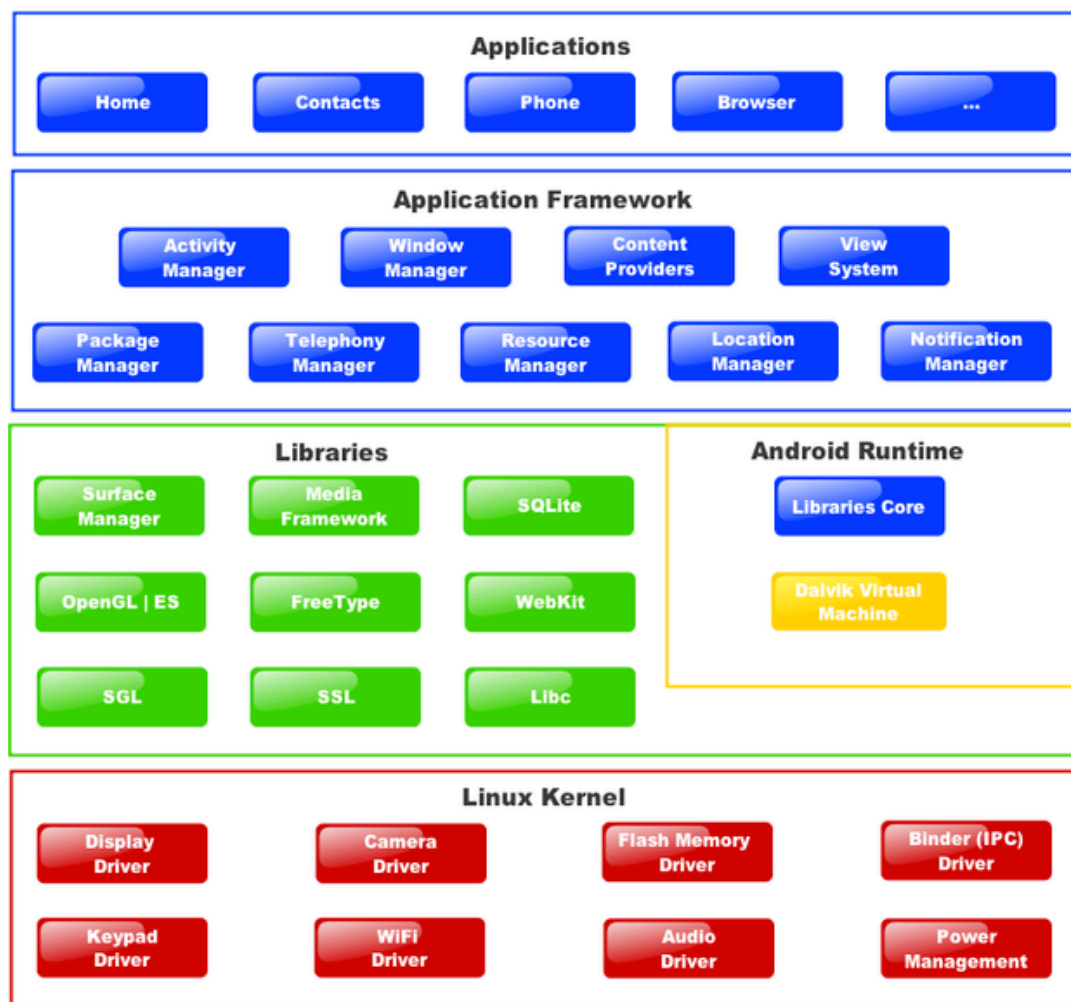


Figura 7. Esquema de l'arquitectura Android.

La Maquina Dalvik és una màquina virtual Java millorada en el rendiment per ajustar-la als recursos d'un dispositiu mòbil. En el moment que es realitza una compilació no es crea arxius *.class* o *.jar* sinó que es crea uns arxius *.dex* (Dalvik Executables) que és el resultat de combinar el codi Java amb les llibreries utilitzades en un o més arxius *.dex* de forma que si alguna part de codi està duplicada es reutilitza reduint el consum de recursos. [1]

4.4. Eclipse.

Eclipse, desenvolupat per Eclipse Foundation, és un entorn de desenvolupament (IDE, Integrated Development Environment) que facilita la implementació d'aplicacions. Està compost per un software base que permet programar en Java al qual se li pot afegir diferents mòduls (plug-in del anglès) de forma que l'usuari adapta el programa a les seves necessitats [2].

Originalment va ser creat per IBM el Novembre de 2001. La fundació Eclipse es creà el Gener de 2004 com a corporació sense ànim de lucre per tal que actués com a administrador de la comunitat Eclipse⁵ i per crear un proveïdor neutral i transparent sobre tot el corresponent a l'aplicació. Avui dia la comunitat Eclipse consisteix en un conjunt transversal d'empreses de la indústria del software.

⁵ <http://www.eclipse.org/>

5. Anàlisi de requisits.

5.1. Implicats.

Alumnes.

Implicats directes en el desenvolupament de l'aplicació. La major part de la població a qui va dirigida l'aplicació pertany a aquest col·lectiu.

Professors.

Juntament amb els alumnes, són també implicats directes en el desenvolupament de l'aplicació. Tot i que no és tan nombrós com el col·lectiu dels alumnes és igual d'important per al correcte desenvolupament del software.

Caps d'estudi i direcció.

Tota planificació d'horaris, professorat i assignatures haurà de tenir en compte actualitzar l'aplicació, per tant s'haurà de comptar amb la col·laboració de caps d'estudi i direcció.

Personal administratiu i serveis (PAS).

Tots els canvis en horaris i terminis de secretaria, així com tots els anuncis s'hauran d'actualitzar en l'aplicació. El PAS s'encarrega d'aquestes tasques.

Dissenyadors i programadors.

Creadors de l'aplicació.

Personal de manteniment informàtic.

Encarregats d'actualitzar l'aplicació cada vegada que hi hagi canvis en documents administratius o acadèmics.

5.2. Usuaris.

Per poder conèixer els usuaris i com actuen es farà una llista dels perfils que s'han detectat i mitjançant la tècnica Personas⁶ es presentaran uns quants exemples de les tasques habituals d'aquests perfils. Això permetrà apreciar la importància de les diferents eines actuals de la pàgina web i les mancances que pot tenir. Juntament amb l'enquesta realitzada, la definició correcta dels usuaris permetrà definir quines funcions es dissenyaran en l'aplicació.

5.2.1. Perfils.

Estudiant.

L'estudiant és, previsiblement, l'usuari més nombrós. Majoritàriament són persones d'ambdós sexes amb edats compreses entre els 18 i els 25 anys. El seu nivell cognitiu en aplicacions informàtiques és alt i per les dades recollides en l'enquesta realitzada a tot el personal de l'EPS sols un 4,3% no disposa de mòbil intel·ligent. Les principals discapacitats que es poden trobar en aquest perfil són degudes a limitacions visuals parcials. Però també pot haver casos de ceguesa total o problemes motrius, tant lleus com severos.

Professor.

Els professors són usuaris amb edats majors de 25 anys. Generalment, el nivell cognitiu en aplicacions informàtiques és alt. A l'igual que els estudiants, les possibles discapacitats que més s'han de tenir en compte són deficiències visuals. De totes formes, sempre es tindran en compte possibles restriccions motrius per qualsevol motiu.

PAS.

PAS correspon a les sigles de "personal administratiu i serveis". La franja d'edat corresponen a aquest perfil és molt ampla i pot ser des dels 16 fins als 67 anys.

⁶ Per a més informació podeu consultar <https://www.uie.com/browse/personas/>

El nivell cognitiu d'aplicacions informàtiques i web pot ser variable segons la funció que realitza. Podríem dir que el nivell general seria mitjà-alt.

5.2.2. Personas.

Estudiant 1.

Francesc Cabús

Nivell informàtic: Mig

Situació: 2on curs del Grau informàtica de gestió

Eines més utilitzades de la web de l'EPS:

- Accés al calendari i horaris
- Correu EPS
- Coneixements:
- Estudis de secundària Batxillerat tecnològic a l'Institut IES Torrefarrera.

Tasques diàries:

Cada dia s'aixeca a les 8 hores per anar a classe a la universitat. Hi és una mitja de 5 hores diàries durant el matí. Per les tardes realitza els treballs necessaris de les diferents assignatures i un parell de dies per setmana, si disposa de temps, aprofita per practicar esport i trobar-se amb els seus amics.

Viu en un pis compartit d'estudiants. No disposa de connexió ADSL al domicili però disposa d'un mòbil intel·ligent amb tarifa plana d'Internet al que utilitza principalment per comunicar-se amb els seus amics. Les consultes del correu o gestions per la web les acostuma a fer durant el matí a l'escola però en alguna ocasió ha tingut que utilitzar el mòbil per realitzar alguna gestió de forma urgent. En alguna d'aquestes ocasions no li ha estat possible realitzar la consulta perquè el mòbil amb que accedeix no és del tot compatible.

Objectius clau en la interacció amb la web de l'EPS:

En Francesc accedeix quasi diàriament al correu de l'EPS durant el matí quan és a l'EPS però si ha d'enviar o rebre algun correu fora de l'horari en que es troba a l'edifici li resulta molt complicat perquè la gestió des d'un dispositiu mòbil és molt enrevessada i lenta.

Habitualment ha de mirar l'horari de classe per saber on té la pròxima classe i en temps d'exàmens no sap mai del cert a quina aula és. Sempre ho té apuntat al calendari del mòbil, però en cada canvi de quadrimestre ha d'introduir l'horari manualment.

Estudiant 2.

Alba López

Nivell informàtic: Alt

Situació: Cursant Màster en Interacció Persona - Ordinador.

Eines mes utilitzades de la web de l'EPS:

- Correu EPS
- Agenda
- Horaris i calendari.

Coneixements:

- Enginyeria tècnica Informàtica de Gestió

Tasques diàries:

L'Alba viu sola a Mollerussa i es lleva cada matí sobre les 6:30 hores per anar a la feina. Treballa en una empresa local de sistemes informàtics com a Assistent en línia del SW que realitza dita empresa. Treballa 8 hores parant 2 hores per dinar tot i que una setmana al mes es queda fins les 22 hores en guàrdia. Normalment, tot i treballar amb un ordinador tot el dia, no té accés a Internet perquè la xarxa de l'empresa no l'hi permet.

Té un dispositiu mòbil amb connexió 3G de tarifa plana i en els descansos pot realitzar alguna consulta si li és necessari. Sobretot l'utilitza per comunicar-se amb la família i els amics. Habitualment també l'utilitza per accedir a continguts multimèdia, premsa i com a navegador GPS. També té un portàtil i connexió ADSL al domicili que és el que utilitza pel màster.

Al finalitzar la jornada laboral a les 18 hores realitza diferents tasques personals i/o domèstiques abans d'anar a sopar. Sempre dedica als estudis l'estona de després de sopar i fins que se'n va a dormir. En les hores de guàrdia, en alguna ocasió s'ha endut el portàtil a la feina per realitzar alguna activitat ja que el volum de feina és molt baix en aquestes hores.

Quan ha d'assistir a alguna classe presencial o als exàmens es pot agafar hores lliures, que recuperarà un altre dia o li descomptaran de les guàrdies, per desplaçar-se a l'EPS. Els dies abans està molt alerta amb els horaris per si hi ha algun canvi en la classe ja que en alguna ocasió ha arribat a l'aula i per diferents motius, no s'ha pogut realitzar.

Objectius clau en la interacció amb la web de l'EPS:

El que més problemes li suposa a l'Alba, tot i que passa poc sovint, és tenir que desplaçar-se a l'escola per qualsevol cosa i no poder-ho realitzar per algun motiu. Aquest desplaçament li suposa, com a mínim, perdre dues hores de feina que haurà de recuperar i un desplaçament de 70km amb vehicle propi.

També li agradaria disposar del correu durant el dia al mòbil ja que sinó no pot accedir al correu fins a finalitzar el dia. Si que el correu és accessible des del mòbil, però el format en que es visualitza li resulta complicat.

5.3. Rols.

Estudiant.

Pertany al rol d'estudiant tot usuari que està matriculat a alguna assignatura. El rol d'estudiant té uns horaris i un calendari específic per a cada usuari. Sol té accés a l'aplicació com a receptor d'informació.

Professor.

Al rol de professor hi pertanyen els usuaris que imparteixen classe en alguna assignatura. Té els mateixos recursos que el rol d'estudiant ja que també té un horari i calendari individual però a diferència d'aquest, el professor accedeix a l'aplicació amb la possibilitat d'introduir o modificar informació en l'àmbit que dirigeix, és a dir, en les assignatures que imparteix.

Administrador de continguts.

El rol d'administrador de continguts és l'usuari que pot modificar tota la informació bàsica de l'aplicació amb l'excepció dels continguts de les assignatures que correspon al rol de professor. Principalment, actualitzarà totes les dades pertanyents a calendaris, normatives i tot lo referent a temes administratius.

Visitant.

A part dels usuaris, que poden desenvolupar un o més dels rols anteriors, també hi haurà usuaris que no en desenvolupin cap d'aquests. És a dir, es considera Visitant qualsevol usuari que s'instal·li l'aplicació i no estigui matriculat a cap assignatura, no imparteixi classe ni s'encarregui d'actualitzar cap tipus d'informació. Aquest rol sol accedirà a la informació bàsica de l'aplicació com serà agenda, notícies, normatives...

5.4. Enquesta sobre els hàbits d'ús de la pàgina web de l'EPS.

En la pàgina web de l'escola hi ha una gran quantitat d'informació però no tot és igualment d'important. Existeixen apartats que ens poden interessar en moments concrets i d'altres que potser només es consulten una sola vegada en tota la carrera. Es necessita saber quins apartats són els més utilitzats pels usuaris i per fer-ho, s'ha realitzat una enquesta a tots els usuaris de l'EPS.

5.4.1. Qüestionari.

El qüestionari està compost de quatre apartats: dades personals, dades del dispositiu mòbil, hàbits d'ús i un apartat de comentaris.

En l'apartat de dades personals ens interessa saber quina titulació està cursant l'alumne perquè en funció del tipus d'estudis que cursa pot ser classes presencials o no i probablement l'ús de la pàgina serà diferent. Inicialment s'havien inclòs unes preguntes sobre l'edat i el sexe de l'enquestat però es van descartar perquè es van considerar innecessàries.

Sobre el dispositiu mòbil interessa conèixer, en cas que disposi de telèfon intel·ligent, el sistema operatiu del seu dispositiu i, en cas que sigui Android, la versió d'aquest. Revisant la pàgina oficial d'Android es trobà que ells realitzen estudis freqüentment sobre els dispositius actius i inclús s'indiquen les dades sobre les mides de pantalla i densitat. Considerant això, es va decidir que no era necessari preguntar-ho als usuaris ja que localitzar el número de versió pot resultar complicat segons el nivell de l'usuari i per altra banda, l'enquesta realitzada oficialment des de Google Play Store amb una mostra de milions de dispositius és molt més fiable que la que es pugui realitzar amb uns centenars de enquestats.

En l'apartat on es consulten els hàbits d'ús es distingeix entre la consulta des d'un PC i el que suposaria l'accés des d'un dispositiu mòbil. La consulta a la pàgina web des d'un mòbil acostuma a ser una consulta que requereix

immediatesa. Podrà ser una consulta a l'horari, per saber l'aula on es realitza una conferència a la que es vol assistir, saber si han sortit unes correccions, etc. Per tant, s'intenta que els usuaris distingeixin entre una cosa i l'altra ja que alguns apartats probablement no siguin necessaris amb l'accés des d'un mòbil. Per finalitzar, es demana que es valorin algunes de les idees que podrien ser útils com pot ser la integració de les assignatures que s'està cursant o el coneixement d'incidències en temps real en qualsevol assignatura.

5.4.2. Qüestionari final.

El qüestionari complert és consultable en l'Annex I (pàgina 127) d'aquest document. Per realitzar la consulta als usuaris es crea un formulari amb Google Drive amb les preguntes corresponents i es sol·licita l'enviament de l'enllaç a través de la llista de distribució de la EPS.

5.4.3. Resultats.

Per a la visualització de les taules de resultats i les gràfiques consultar l'Annex II (pàgina 133). S'han obtingut 183 respostes en 2 mesos aproximadament.

5.4.4. Anàlisi dels resultats.

Fent una lectura general dels resultats de l'aplicació es treuen les següents conclusions.

- I. Un 71% dels usuaris que disposen de telèfon intel·ligent utilitzen un dispositiu que funciona amb Android. Aquesta dada ens serà de vital importància en el moment de seleccionar en quines plataformes es desenvolupa l'aplicació.
- II. Principalment els usuaris utilitzen el telèfon per a tres coses. Missatgeria instantània (78%), consulta de correu electrònic (50%) i consulta de les xarxes socials (40%). Això indica que principalment l'ús que es dona als dispositius portàtils, amb capacitat de realitzar infinitat de funcions, és

activitats que ens requireixen una resposta ràpida. Per tant, en la nostra aplicació ens hem de centrar també en la informació que requireix aquesta propietat.

- III. En les preguntes 5 i 6 els usuaris ens indiquen les seccions de la pàgina que visiten freqüentment i quines creuen necessàries en el dispositiu mòbil. Queda clar que "Horaris i Calendari" és una secció imprescindible seguit de les seccions "Matrícula", "Pràctiques", "Guies Docents" i "Agenda". "Plans d'estudi" i "Altres tràmits de matrícula" són dues seccions que també es consideren importants per més del 40% d'enquestats però en el cas de "tràmits de matrícula", en la pregunta 5, es reflexa que realment molt poca gent hi accedeix i en el cas de "plans d'estudi", revisant la pàgina web, s'observa que ens porten als mateixos documents.
- IV. El cas de la secció de pràctiques és curiós. Més del 50% creu que s'hauria d'incloure en l'aplicació aquesta secció però si revisem la web no hi ha pràcticament informació sobre el tema. En el moment que es realitza aquest estudi solament es pot consultar una llista de les empreses col·laboradores i omplir un formulari per sol·licitar un conveni de pràctiques en cas que siguis una empresa. Aquest fet revela que hi ha un fort interès en aquest tema i que la pàgina web probablement no els hi aporti la suficient informació.
- V. De les propostes que s'han realitzat per afegir a l'aplicació, la més valorada és el punt de comunicació d'incidències en el moment. Això ens enllaça amb el punt II d'aquest anàlisi. Excepte la possibilitat de que en el calendari es disposés del temari que es tractarà en l'esdeveniment, la resta de propostes són bastant ben valorades.
- VI. En l'apartat de Propostes i comentaris destaca, per sobre de tot, la integració del Campus Virtual o que almenys la informació que es mostra en l'aplicació fos sincronitzada amb aquest. També es proposa poder configurar el correu dels alumnes en el dispositiu i enviament de notifikacions amb les notes en temps real.

5.5. Anàlisi contextual de tasques.

Analitzant el resultat de l'enquesta s'han determinat les tasques que es necessari que pugui realitzar l'aplicació. Es vol que l'aplicació serveixi per accedir en els punts més importants de forma ràpida però sense fer que sigui massa "gran". No es vol tenir totes les funcions de la pàgina web per què això suposa crear una aplicació que ocupa espai, requereix més recursos i no es centra tant amb les tasques que es creu que són necessàries en un dispositiu mòbil.

S'han separat les tasques en funció del resultat de l'enquesta. S'han classificat com a tasques principals tots els punts que en l'enquesta almenys la meitat dels usuaris els han considerat " importants " o " imprescindibles " i com a tasques opcionals els punts que majoritàriament ho han considerat almenys " útil ".

5.5.1. Tasques principals.

- Consulta de l'agenda d'activitats de l'EPS.
- Consulta de l'horari de classe d'una de les titulacions de l'EPS.
- Consulta del calendari acadèmic d'una de les titulacions de l'EPS.
- Consulta del calendari d'exàmens d'una de les titulacions de l'EPS.
- Consulta de la guia docent d'una de les titulacions de l'EPS.
- Consulta dels tràmits necessaris per sol·licitar pràctiques en una empresa.
- Consulta del calendari de matriculació de titulacions i TFC/PFC.
- Consulta de la documentació necessària per matricular-se.
- Consulta dels tràmits necessaris per realitzar l'auto-matricula.
- Consulta d'informació dels Màsters que s'imparteixen en l'EPS.

5.5.2. Tasques opcionals.

- Consulta de notícies recents en l'EPS.
- Consulta el pla d'estudi d'una de les titulacions.

- Consulta dels terminis de modificació, ampliació i anul·lació de matrícula.
- Consulta dels terminis de renúncia d'assignatures.
- Consulta dels terminis de reconeixement de lliure elecció o matèries transversals.
- Consulta de les ofertes de lliure elecció i matèries transversals.

5.5.3. Anàlisi de "la competència".

Per analitzar aplicacions similars s'han buscat un rànquing de les universitats més importants del país per seleccionar una mostra objectiva i s'ha trobat un article de Expansion.com⁷ del maig de 2014 basat en un estudi de la Fundació BBVA-live. S'ha descarregat l'aplicació que tenen, si és que en tenen, i s'ha revisat uns aspectes bàsics que serviran de guia.

Mida de l'instal·lador: Volem crear una aplicació que no consumeixi gaire espai en memòria però ens anirà bé saber quina és la mida mitja d'aquest tipus d'aplicacions.

Utilitats: Resum de les funcions que proporciona l'aplicació.

Estructura: Breu descripció de l'estructura de navegació de l'aplicació.

Aparença: Comentari sobre l'estructura visual de l'aplicació.

Comentari: Valoració de l'aplicació i dels punts comentats anteriorment.

Universidad Complutense de Madrid. (Figura 8)

Mida de l'instal·lador: 3,68 MB

Utilitats: L'aplicació ens permet accedir a tota la informació referent a la universitat. Titulacions, centres, professors, mapa i contacte. Té un accés, també, a la biblioteca, xarxes socials del centre i notícies.

⁷ <http://www.expansion.com/2014/05/14/entorno/1400069541.html>

Estructura: La activitat d'inici és la secció de Notícies. La navegació es realitza per una configuració de pestanyes a la part inferior de i totes les seccions tenen estructura de llista.

Aparença: Fons blanc amb marca d'aigua del emblema de universitat, Icones i text amb vermell i els subtítols en blau. Tots els elements de les llistes estan formades per icona i text.

Comentari: Aplicació bastant ràpida de moviments tot i que per arribar a segons quin punt d'informació pots necessitar fins a 5 moviments. La tipografia del text i el color vermell o blau dificulten lleugerament la lectura.



Figura 8. Aplicació de la Universitat Complutense de Madrid.

Universitat Politècnica de València (Figura 9)



Figura 9. Aplicació de la Universitat Politècnica de Valencia.

Mida de l'instal·lador: 4,18 MB

Utilitats: Inici en la secció de Notícies. Disposa d'informació de radio/TV de la universitat, agenda que es pot personalitzar si s'inicia sessió, mapa dels punts importants, accés al correu i a la intranet però al no tenir credencials per accedir no podem saber més. En l'apartat de configuració ens permet canviar d'idioma entre Castellà i Valencià.

Estructura: L'aplicació s'inicia en l'apartat de Notícies i la navegació entre seccions es realitza mitjançant un menú desplegable lateral.

Aparença: Lectura clara i còmoda. Entorn visual senzill i agradable.

Comentari: Es considera que de les estudiades és la més semblant al que es voldria realitzar per aquest projecte. Ràpida d'executar i de moviments senzills i àgils.

Universitat de València (Figura 10)

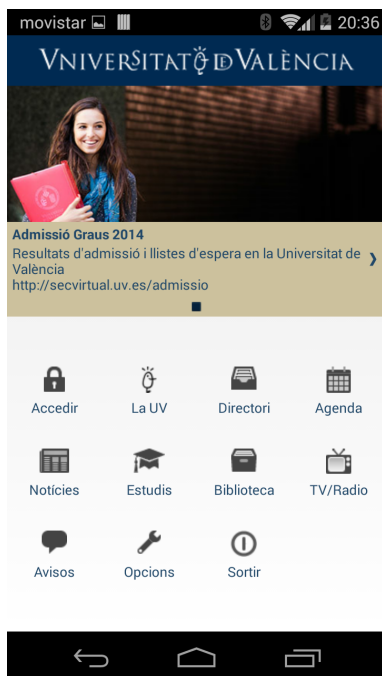


Figura 10. Aplicació de la Universitat de València.

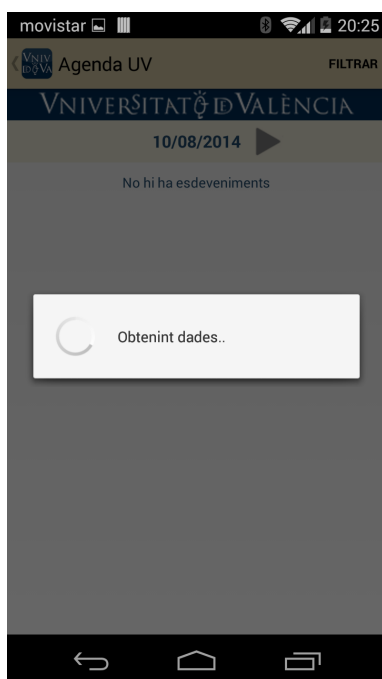


Figura 11. Captura del moment en que l'aplicació de la UV realitza la carrega de dades.

Mida de l'instal·lador: 5,85 MB

Utilitats: Accés a la informació bàsica com el directori, agenda, notícies, estudis, TV/radio i biblioteca. A part d'això hi ha un accés privat que requereix tenir credencials, que per la informació que et dona Google Play al descarregar l'aplicació, et dona accés a les assignatures matriculades, horaris i notificacions de notes. També hi ha una àrea d'avisos que ens informa de qualsevol incidència.

Estructura: A l'iniciar l'aplicació s'accedeix a la pàgina central de menú. Des d'aquí s'accedeix a qualsevol secció. Per navegar entre seccions sol tens l'opció de tornar enrere i accedir a través del menú.

Aparença: Text i icones es distingeixen correctament però de mida una mica reduït. Els colors blau fosc, cru i blanc combinen correctament per fer un entorn visualment atractiu.

Comentari: La meitat de la pàgina inicial de l'aplicació està ocupada per una imatge i una franja amb una informació a destacar. Es podria dir que, per aquest motiu, les icones i el text queden d'una mida molt reduïda però veient que tot i això haguessin pogut ampliar-los una mica més sense

sobrecarregar-ho gens, s'entén que la línia de disseny és aquesta. Text i icones de mida massa reduïda. Com es deu veure en un dispositiu amb pantalla més petita? (El meu és de 4,5"). També s'ha detectat que en la secció d'agenda es tarda en obtenir tota la informació i es queda amb un diàleg de càrrega durant uns segons, depenent de la connexió. Cada vegada que s'entra la realitza de nou, cosa que dona a entendre que no es guarda res en el dispositiu (Figura 11).

Universitat de Barcelona i Universitat Autònoma de Barcelona (¡Error! No se encuentra el origen de la referencia. i Figura)

Aplicació que únicament et permet rebre notificacions de notes, revisions d'exàmens, etc. en el moment que s'introdueixen al teu expedient. Requereix credencials de la universitat i no es pot avaluar el funcionament. Són aplicacions dedicades a un propòsit que, es suposa, es compleix.



Figura 13. Aplicació de la Universitat de Barcelona.

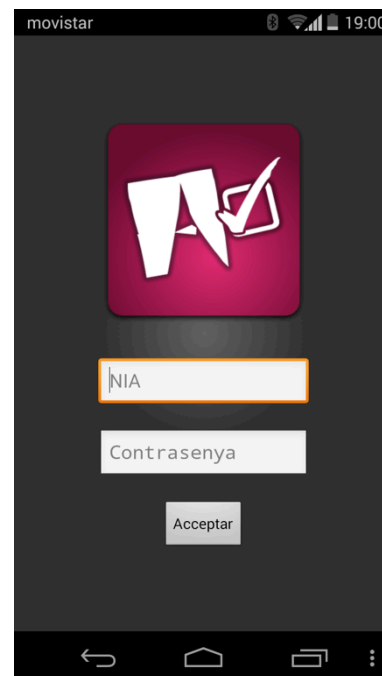


Figura 12. Aplicació de la Universitat Autònoma de Barcelona.

Universitat Politècnica de Catalunya. (Figura 14)

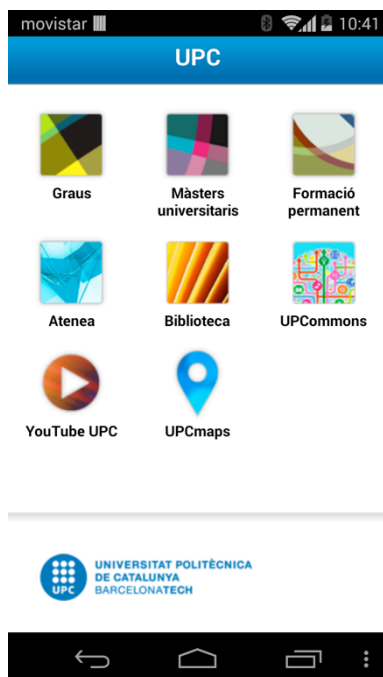


Figura 14. Aplicació de la Universitat Politècnica de Catalunya.

Mida de l'instal·lador: 537KB

Utilitats: Accés a guies docents de tots els estudis que imparteixen, biblioteca, campus virtual, canal de Youtube i ubicació de tots els edificis de la universitat.

Estructura: Està format per un menú d'accés en forma de graella. Únicament aquesta interfície es càrrega a l'instal·lar l'aplicació ja que en el moment que s'entra en qualsevol secció és una navegació WebView. Per desplaçar-se entre seccions s'ha de tornar enrere amb el botó del mòbil fins al menú inicial.

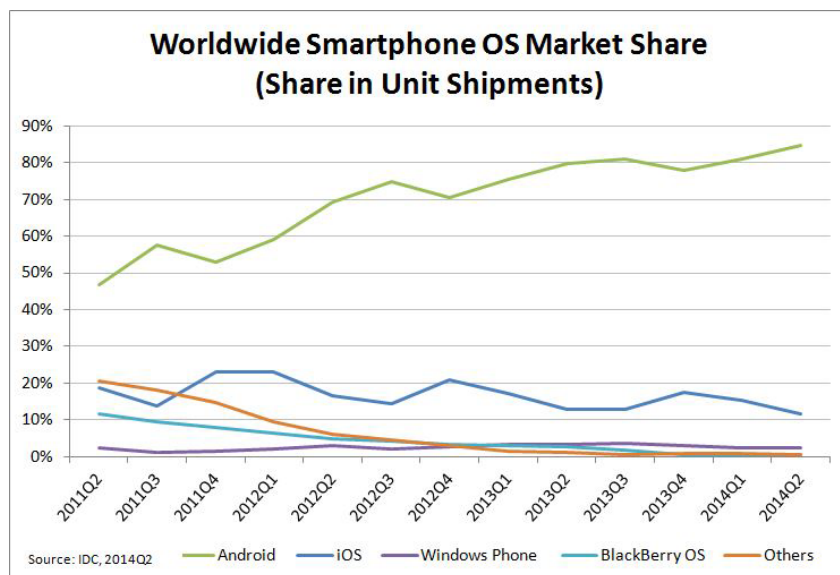
Aparença: Les icones són un conjunt de colors visualment correctes però que, a excepció del canal YouTube i el mapa que si que aporten informació a través de la imatge, la resta no aporten cap tipus d'informació sobre on ens condueixen.

Comentari: Visualment molt correcta ja sigui a nivell cromàtic com de dimensions i en les pàgines interiors es manté el mateix esquema visual. La navegació tipus web en les pàgines internes provoca lentitud a l'hora de carregar cada secció però implica que l'aplicació solament ocupa instal·lada 1,7MB.

5.6. Plataforma.

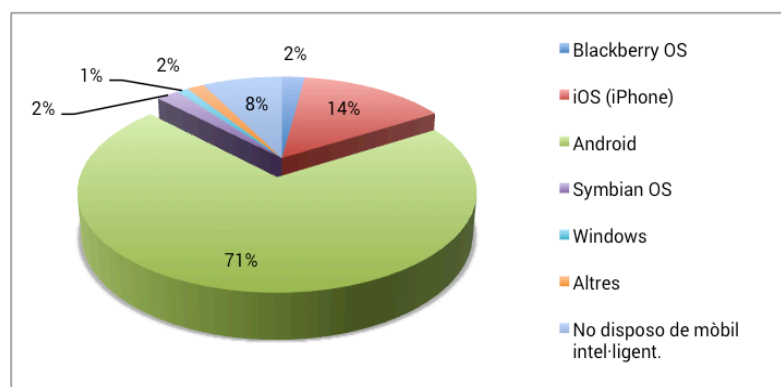
5.6.1. Selecció de Plataforma.

El títol del projecte ja indica que la plataforma amb que es programa l'aplicació és Android però aquesta decisió no ha estat aleatòria. Per motius de calendari es preveu implementar-ho en una sola plataforma. En diferents estudis que es realitzen anualment queda clar que l'evolució de ventes de dispositius mòbils amb sistema operatiu Android va en augment. En el Gràfic 2, realitzada per la IDC⁸ (Internacional Data Corporation), es mostra l'ascens de ventes de terminals en els últims anys a nivell mundial.



Gràfic 2. Evolució de venta de dispositius mòbils.

També en l'enquesta realitzada en l'Anàlisi de Requisits queda reflectida la diferència d'Android sobre la resta de sistemes.



Gràfic 3. Respostes obtingudes de la Enquesta als usuaris de l'EPS.

⁸ <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>

5.6.2. Versió.

En la mateixa pàgina de *Android Developers*⁹ es pot veure les estadístiques obtingudes per l'aplicació *Google Play Store*¹⁰, que és el portal de compra de Android i Google. Les estadístiques s'obtenen a partir dels dispositius que tenen instal·lada l'aplicació, que són tots excepte els inferiors a la versió 2.2, i registra les dades dels que accedeixen al portal en els últims set dies per la qual cosa són bastant actualitzades.

Els dispositius que funcionen amb la versió de Android inferior a la 2.2 no suporten l'aplicació *Google Play Store* per tant, no estan comptabilitzats en les estadístiques. Tot i això, s'indica en la web que ja en l'Agost de 2013 els dispositius signats en els servidors de Google eren menys del 1%.

Durant la realització d'aquest projecte s'ha consultat diverses vegades aquesta pàgina. Amb les dues gràfiques obtingudes es pot veure la progressió que segueix. Com és d'esperar els dispositius antics minvaran mentre que aniran sortint versions noves de Android i ocuparan més quota.

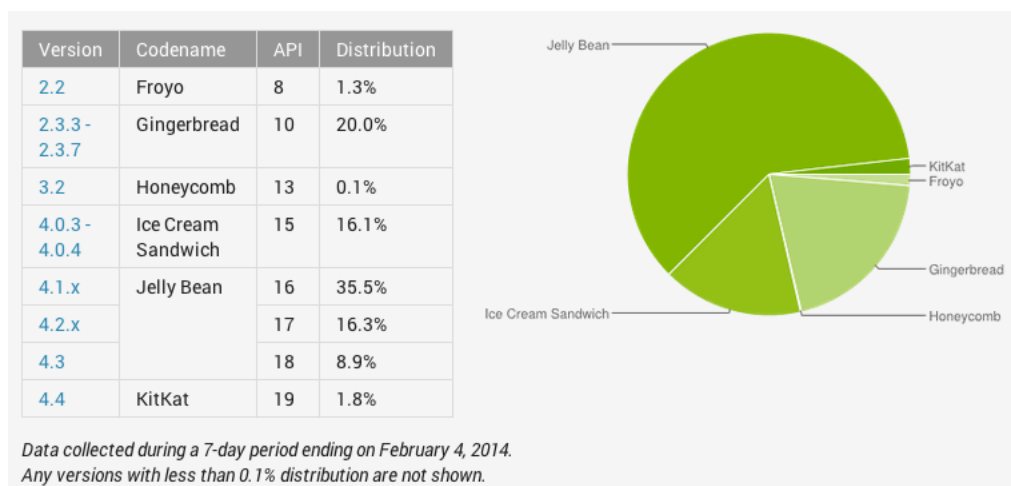


Figura 15. Estudi de Versions de Google en els 7 primers dies de Febrer de 2014.

⁹ https://developer.android.com/about/dashboards/index.html?utm_source=ausdroid.net

¹⁰ <https://play.google.com/store>

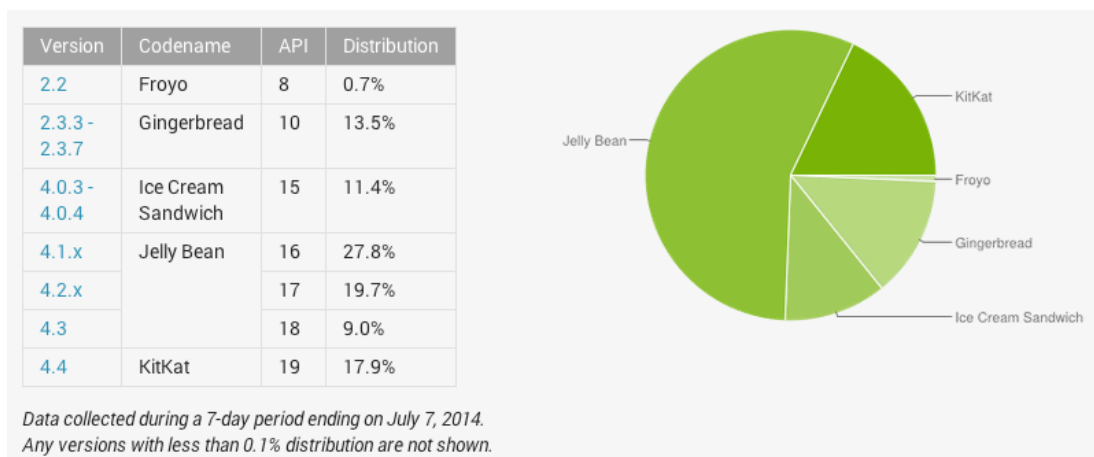


Figura 16. Estudi de Versions de Google en els 7 primers dies de Juliol de 2014.

La versió important a tenir en compte és la 3.0. A partir d'aquesta versió és compatible l' ActionBar i es crea la programació amb fragments entre altres coses. Aquest segon element finalment no s'utilitzarà per implementar l'aplicació però sí l' ActionBar tal com s'explica en el punt 8.2.7 del document. Segons les dades proporcionades per Google, al febrer de 2014 hi havia un 21,3% de dispositius amb versió inferior però ja al juliol era el 14,2%. Es pot implementar de forma que a partir de la versió 2.1 siguin compatibles però per problemes de calendari sol s'implementarà a partir del 3.0.

5.6.3. Pantalla

D'igual forma que la versió, Google guarda les dades referents a mides de la pantalla i densitat d'aquests dispositius. Això és important per definir les dimensions de les icones, dibuixos i distribucions de pantalla, ja que podria donar-se el cas que en un dispositiu de pantalla gran o molt gran la interfície es visualitzés correctament però en reduir-ho a una pantalla petita o de baixa densitat es visualitzés molt petit o canviessin les posicions.

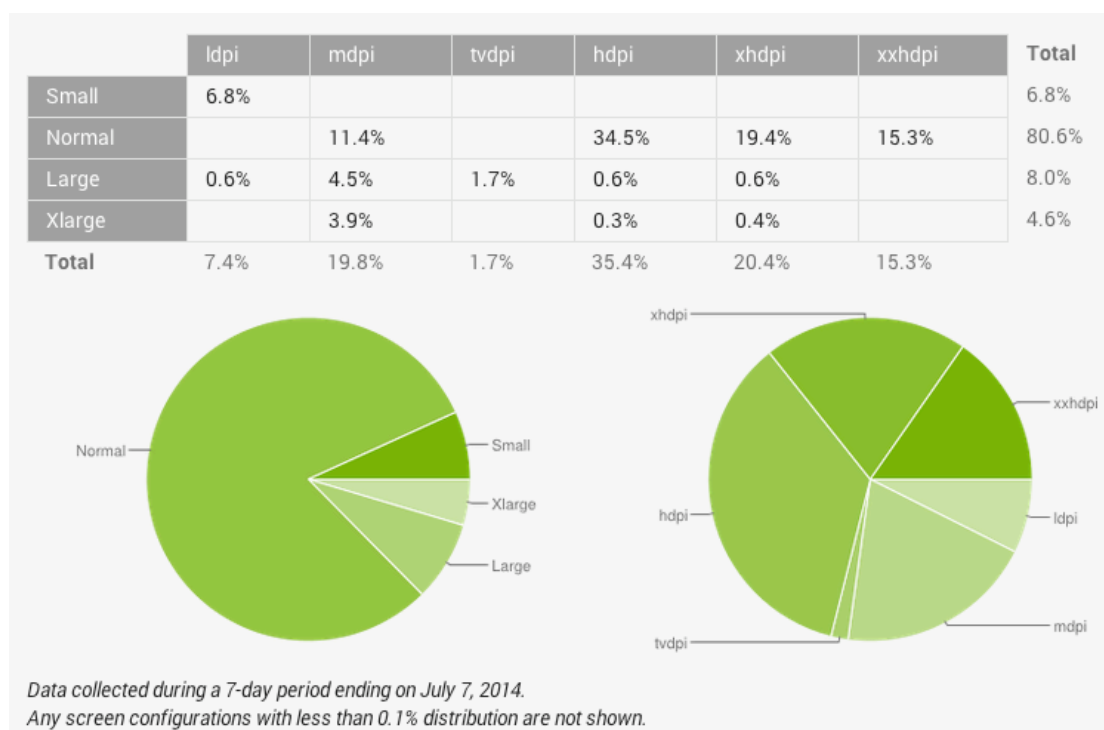


Figura 18. Estudi de mida i densitat de pantalles realitzat per Google els 7 primers dies de juliol de 2014.

S'ha de tenir en compte que les estadístiques que ofereixen no sol inclouen els anomenats "dispositius petits" com són els mòbils, sinó que també inclou tauletes. L'adaptació de les mesures que indiquen no és exacta com es pot veure en la Figura 17.

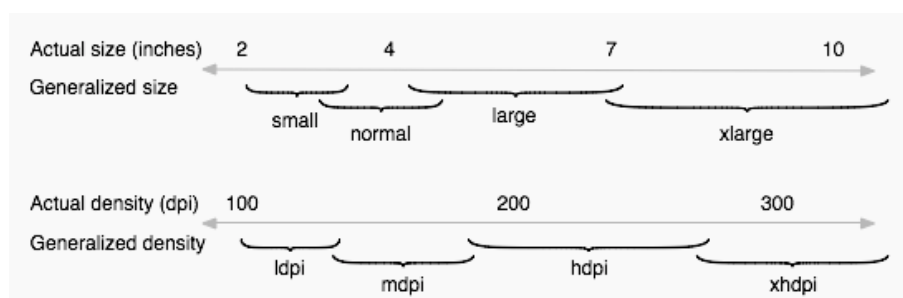


Figura 17. Relació entre les etiquetes i els valors de la mida de pantalla i densitat.

Aquest projecte s'ha centrat en dispositius mòbils per tant, es deixa de banda la mida extra gran (xlarge). La mida petita amb la densitat LDPI tampoc s'utilitza però el disseny d'icones i mides de les imatges per aquesta densitat és la mateixa que per MDPI i el mateix dispositiu fa l'escalat. Per tant, si tenim en compte dissenyar de forma que la mida sigui flexible, no ha de donar cap problema la utilització per aquests dispositius.

5.7. Objectius.

5.7.1. Objectius funcionals.

Espai en memòria.

Volem crear una aplicació que no ocupi gaire espai. En l'anàlisi de la competència s'ha vist que en general l'instal·lador de les aplicacions ocupa entre 3 i 6 MB. S'ha d'intentar que l'aplicació de l'EPS no ocupi més de 6 MB però a ser possible que ocupi el mínim d'espai.

Rapidesa d'execució.

L'aplicació ha de ser àgil i que en cap moment et faci esperar. L'analista de usabilitat Jacob Nielsen¹¹ va definir al 1993 uns límits de temps de reacció de les pàgines web però que són perfectament vàlids avui dia i també per dispositius mòbils. Els temps de resposta els divideix en tres intervals:

- 0,1 segons. En que l'usuari té la sensació que la resposta és instantània.
- 1 segon. Tot el que sigui menys d'un segon l'usuari ja percep que hi ha un temps d'espera però accepta que és el temps que triga el sistema en generar la resposta i encara es manté concentrat en la tasca que està realitzant.
- 10 segons. En l'interval de 1 a 10 segons l'usuari percep el retard amb disgust però encara pot mantenir la concentració en la tasca.
- Tot el que sigui més de 10 segons, segons Nielsen, implica que l'usuari ja comença a pensar en d'altres coses abans de rebre la resposta del sistema. Percep que no disposa del control de les accions ja que és el dispositiu qui marca els temps.

Tenint en compte això, és necessari que totes les accions que realitzem dins de l'aplicació i que no suposin una descàrrega de dades entrin en l'interval de 0,1 a 1 segons de retard. En cas que s'hagi de realitzar alguna descàrrega com pot ser

¹¹ <http://www.nngroup.com/articles/website-response-times/>

les actualitzacions de l'agenda o les notícies ja dependrà de la connexió per la qual cosa és recomanable que es realitzin sempre en segon pla.

S'ha de dir que no es disposa de la possibilitat de mesurar els temps de resposta en dècimes de segon i no es podrà confirmar mecànicament aquest requisit.

Consum de dades.

Actualment les tarifes de les operadores per consum de dades mòbils, que tenen limitacions en el volum de descàrrega, són bastant més costoses econòmicament que les de línies fixes. L'aplicació ha d'oferir la possibilitat a l'usuari de gestionar les connexions i descarregues. Això es podria fer de forma que l'usuari tries el moment en que vol que es descarregui el contingut o donar l'opció de seleccionar el tipus de xarxa preferida per fer la descàrrega. Les opcions serien descarregar solament a través de connexions sense fils (Wi-Fi) o realitzar-la en qualsevol moment a través de qualsevol xarxa, que suposarà la connexió amb Wi-Fi o banda ampla mòbil.

5.7.2. Objectius d'usabilitat.

Aprenentatge.

L'aplicació ha de ser fàcilment intel·ligible. El primer cop que utilitzin l'aplicació han de saber arribar a la informació que desitgen de forma fàcil i sense massa errors.

Agilitat de navegació.

Majoritàriament els dispositius Android són aparells amb pantalla tàctil. Per tant, s'intentarà minimitzar la utilització del teclat per tal de que la navegació sigui àgil i ràpida.

Estètica.

Es procurarà crear un entorn agradable al treball i clar. S'intentarà mantenir l'esquema cromàtic de la pàgina web per què afavoreix la identificació de l'aplicació.

Càrrega cognitiva.

L'usuari ha de ser capaç d'arribar a la informació desitjada en un màxim de 3 clics des de l'inici de l'aplicació. Es defineixen 3 clics per què es té en compte que iniciem des d'un menú que ens permet l'accés directe a la secció que volem. Seria bo que es pugues donar l'opció d'iniciar l'aplicació en la secció que desitges l'usuari, ja que és possible que una de les seccions sigui la més visitada i ho prefereixi.

Recuperabilitat.

Hem de tenir en compte que els usuaris poden equivocar-se i s'ha de preveure l'opció de recuperar les dades introduïdes o de tornar al punt on era amb facilitat.

5.7.3. Objectius d'accessibilitat.

Reconeixement de veu.

S'intentarà habilitar un sistema per poder navegar per l'aplicació mitjançant la veu, ja sigui a través del menú com per escriure on es requereixi.

Mida de fonts i icones.

L'aplicació ha de poder ser visible en almenys dos mides d'icones i lletra. Tot i això, el dispositiu sempre tindrà l'opció d'ampliar el contingut de la pantalla.

Colors.

S'ha d'evitar colors problemàtics per a daltònics. Utilitzar vermell i verd el mínim possible per evitar-ho.

Idioma.

Oferir la possibilitat a l'usuari de seleccionar, a poder ser, el seu idioma. El més bàsic és castellà i si es pot incloure algun idioma més, millor.

6. Disseny.

6.1. Arquitectura de la informació / definició de les seccions.

A partir de les tasques definides per a l'anàlisi de requisits i revisant la informació que es dona a la pàgina web s'han creat unes targetes per iniciar el procés d'ordenació mitjançant la tècnica Card sorting. En les propostes dels enquestats sortien punts com el Campus Virtual i el correu electrònic de l'Escola. No s'han inclòs per la dificultat que comporta la implementació d'aquests punts però es proposaran com a treballs futurs ja que són punts de gran utilitat si es realitza correctament.

La tècnica Card sorting consisteix en dividir la informació que tenim per la nostra aplicació en targetes, tipus Post-it per exemple, i deixar que diferents usuaris realitzin la distribució com creguin més entenedor. L'usuari agruparà les targetes segons ell cregui que són d'un tema similar o que tenen relació. Es proporcionarà targetes extra i retolador per escriure el títol d'un grup en una targeta i si creu que algun temari pugui estar separat o junt és lliure de organitzar-ho com vulgui. Quan es tenen varies versions es valora la que s'adoptarà definitivament¹².

Per problemes de calendari solament s'ha realitzat una versió de Card sorting.

¹² Per més informació consultar: <http://www.grihotools.udl.cat/mpiua/ordenacion-de-targetas-card-sorting/>

6.1.1. Card sorting.

Les targetes que s'han preparat són les següents:

- Notícies
- Agenda
- Guia docent (sol l'actual)
- Horaris de classe
- Calendari acadèmic
- Calendari exàmens
- Calendari de matriculació
- Practiques tutelades en empresa
- Convenis de pràctiques en empresa
- Documentació necessària per la matrícula
- Preus de matrícula
- Oferta d'assignatures
- Més informació de matrícules de Graus
- Més informació de matrícules de Màster
- Modificació / ampliació de matrícula
- Anul·lació de matrícula
- Renúncia d'assignatures
- Matrícula de pràctiques tutelades
- Matrícula de treball integrat en un grup de recerca
- Matrícula de TFC/PFC
- Reconeixement Lliure elecció / Matèria transversal

Resultat de l'ordenació d'aquestes targetes.

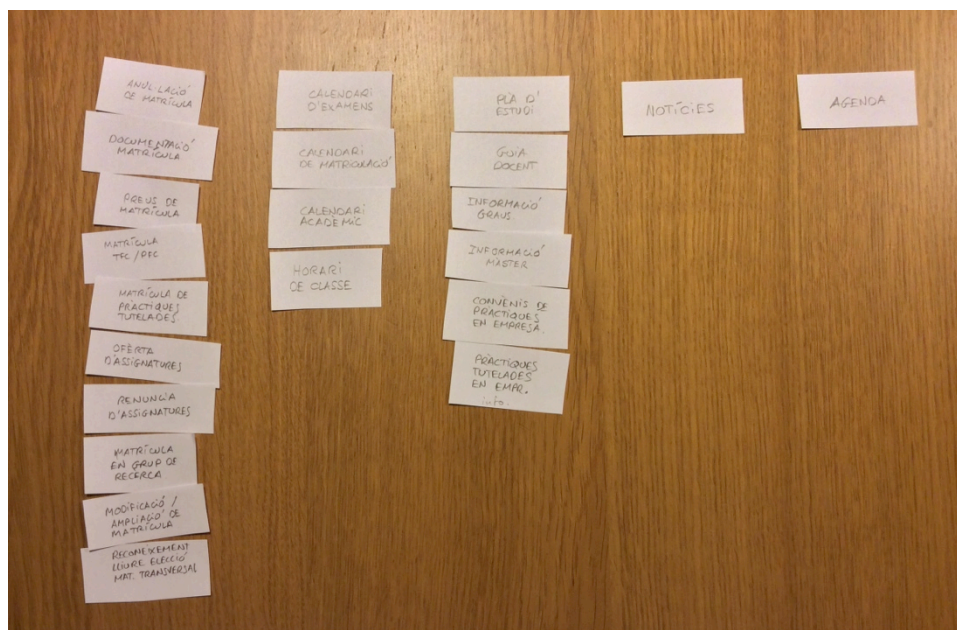


Figura 19. Resultat de la realització del Card Sorting.

Columna 1. Matrícula.

- Anul·lació de matrícula.
- Documentació Matrícula.
- Preus de Matrícula.
- Matrícula de Practiques tutelades.
- Oferta d'assignatures.
- Renuncia d'assignatures.
- Matrícula en grup de Recerca.
- Modificació / Ampliació de Matrícula.
- Reconeixement de Lliure Elecció, Mat. Transversal.

Columna 2. Calendari.

- Calendari d'exàmens.
- Calendari de matriculació.
- Calendari acadèmic.
- Horari de classe.

Columna 3. Guia Docent.

- Pla d'estudi.
- Guia Docent
- Informació de Graus.
- Informació de Màster.
- Convenis de pràctiques en Empresa.
- Pràctiques tutelades en Empresa.

Columna 4. Notícies.

- Notícies.

Columna 5. Agenda.

- Agenda.

6.1.2. Estructura definitiva.

Amb la realització del Card sorting s'ha decidit la distribució definitiva. Finalment s'han creat 5 seccions separades per activitats diferents.

- Agenda, on es mostrarà els esdeveniments programats futurs del blog de l'EPS.
- Notícies, on es mostrarà totes les notícies ja passades del blog de l'EPS.
- Matrícula, on s'inclourà els enllaços als punts de més interès de l'apartat de Matrícula de la web de l'EPS.
- Guia docent, on s'inclourà els enllaços a les guies docents de les diferents titulacions que ofereix l'escola. Sol s'inclourà les de l'any vigent ja que sempre es pot accedir a través de la pàgina web si és necessari algun altre any.
- Calendari, on es mostrarà les classes, festius i períodes importants per als alumnes de forma individualitzada.

6.2. Prototip de paper¹³.

El primer prototip que es realitza és a mà alçada i amb llapis. Després de diferents esbossos (Figura 20) creem un primer prototip de paper complet (Figura 21, Figura 22 i Figura 23).

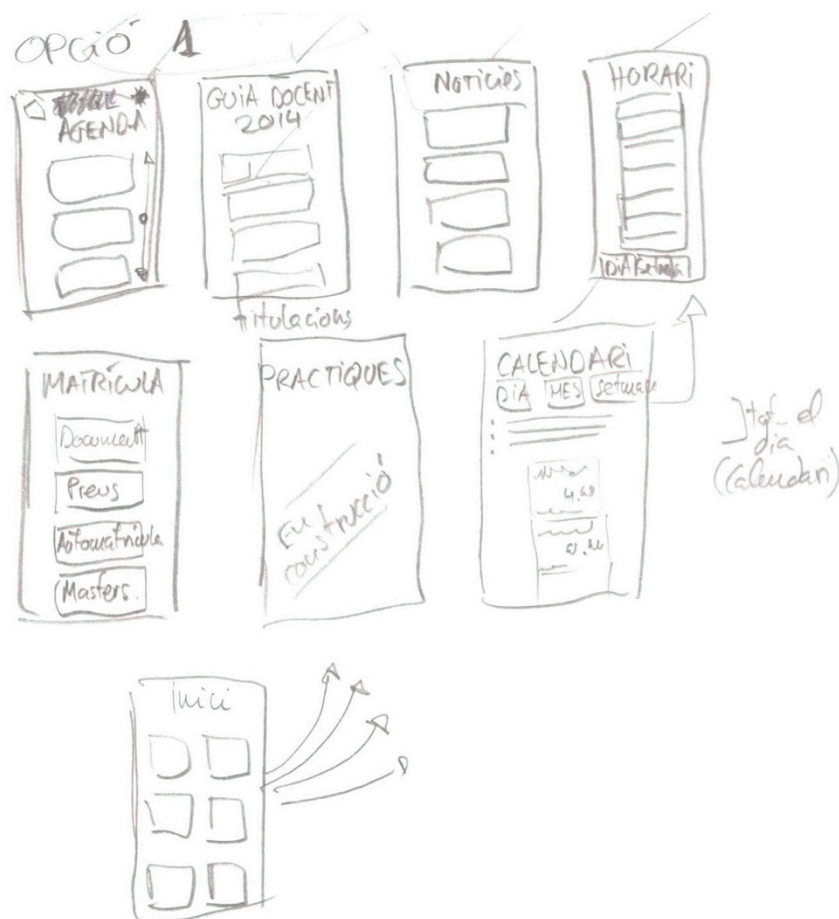


Figura 20. Primer esbós de les seccions abans de crear el prototip de paper.

¹³ Per a més informació visiteu: <http://www.grihotoools.udl.cat/mpiuu/prototipos-de-papel-paper-prototyping/>

PROTOTIP DE PAPER 1.0

pàgina 1

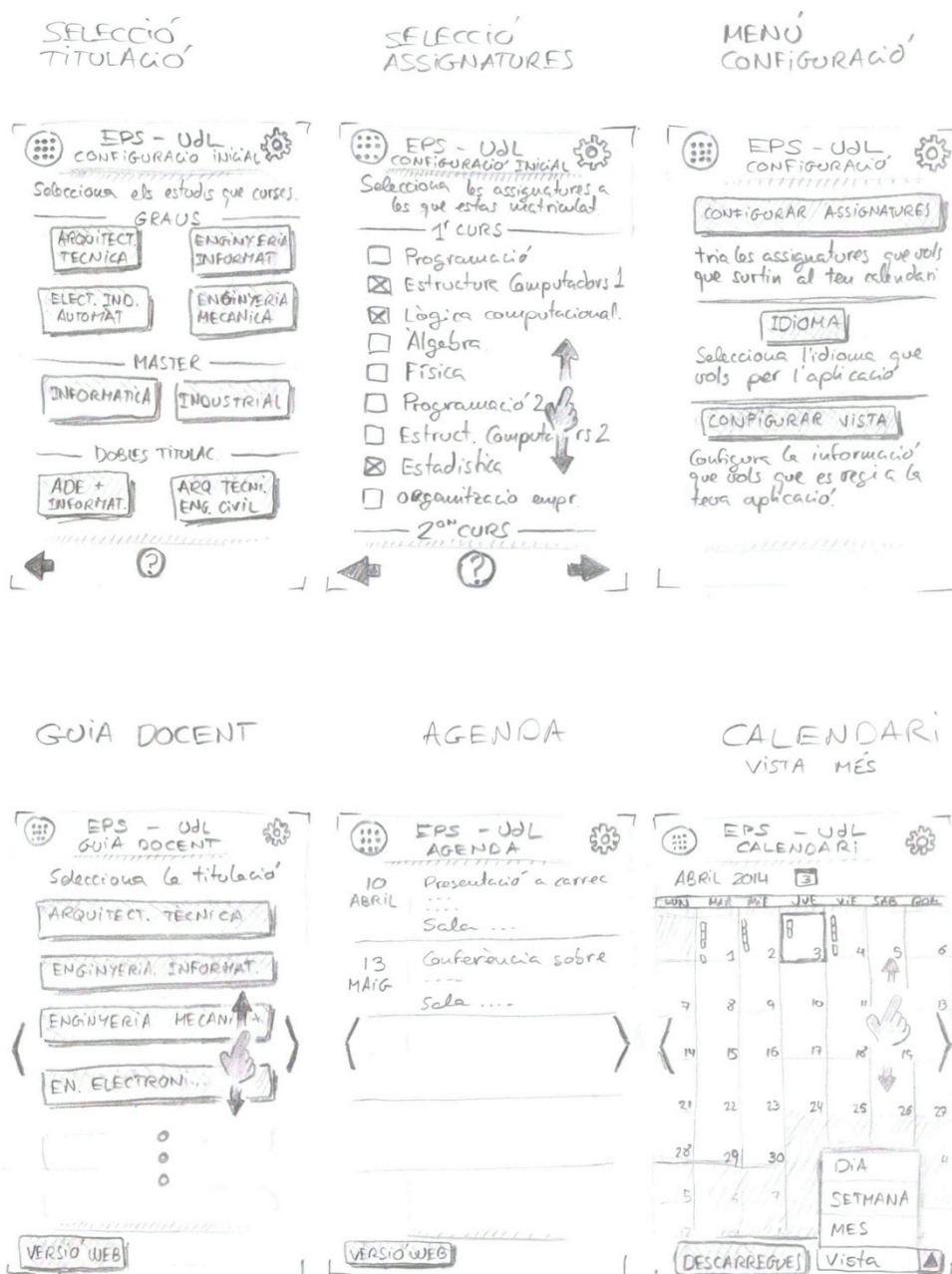
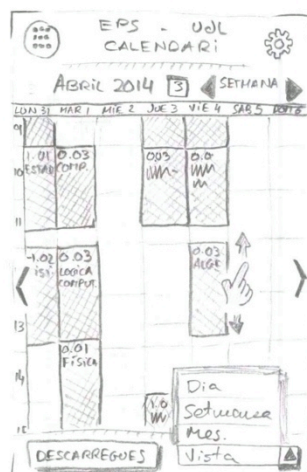


Figura 21. Prototip de paper versió 1, pàgina 1

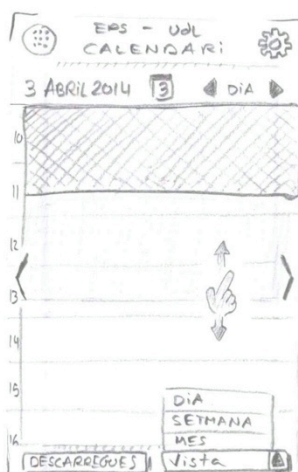
PROTOTIP DE PAPER 51.0

pàgina 2

CALENDARI VISTA SETHANA



CALENDARI VISTA DIA



CALENDARI DESCARREGUES



NOTÍCIES



Figura 22. Prototip de paper versió 1, pàgina 2

PROTOTIP DE PAPER V1.0

pàgina 3

CONFIGURAR VISTA (1)



PAGINA INICI • MENÚ.

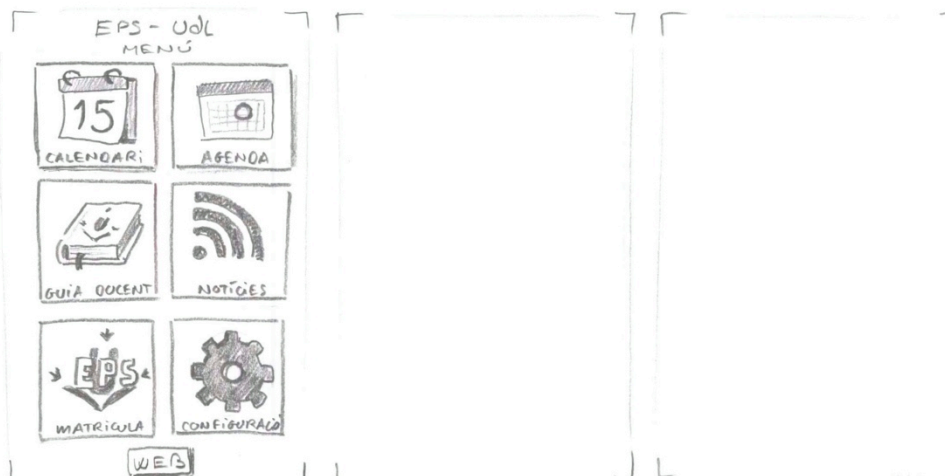


Figura 23. Prototip de paper versió 1, pàgina 3

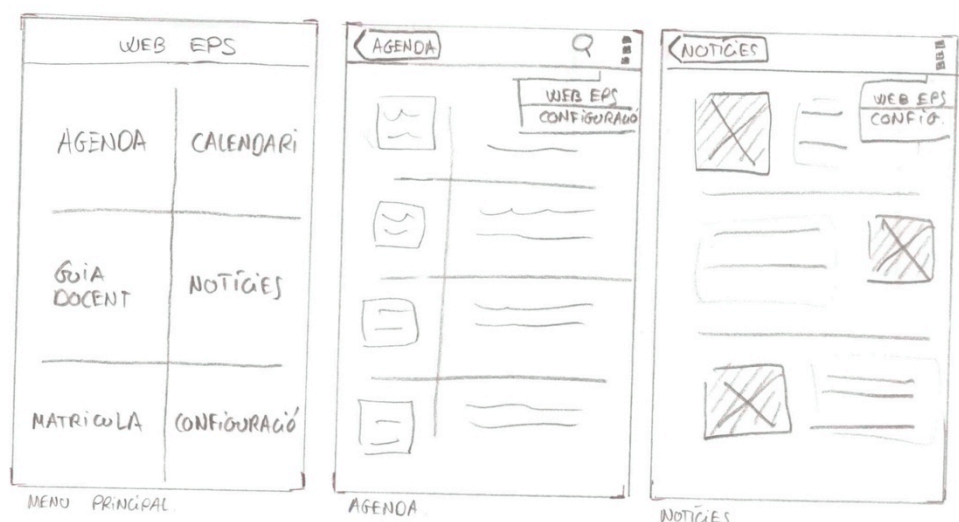
Després de crear el primer prototip de paper i quan s'inicia la creació del prototip digital es comença a recollir informació sobre la distribució i aparença de les aplicacions Android bàsiques. Veient que el format és bastant diferent es crea una nova versió actualitzant aparença, icones i afegint elements que en un primer prototip faltaven (Figura 24, Figura 25 i Figura 26).

En aquest segon prototip s'ha anotat en cada secció les possibles tasques que es poden realitzar en el test amb els usuaris.

Amb el disseny del prototip de paper ha sortit el dubte de si és necessari que l'aplicació disposi d'una interfície "menú" o no. Donada la mida de l'aplicació, que sol compta amb cinc seccions més la d'ajustos, es pot entendre que la navegació d'una secció a una altra és molt ràpida, ja que es pot fer desplaçant el dit horitzontalment per la pantalla les vegades que siguin necessàries. L'usuari podria configurar la secció que ell prefereixi com a inicial i a partir d'aquí desplaçar-se lateralment on desitgi. L'altra opció és que l'aplicació arrenqui directament a una pantalla on tinguem l'enllaç directe a cada secció (menú), també la de ajustaments, mitjançant un botó. Si és necessari, tal com s'indica l'apartat d'objectius d'usabilitat de l'anàlisi de requisits, afegir l'opció que l'usuari canviï la secció d'inici per la desitjada.

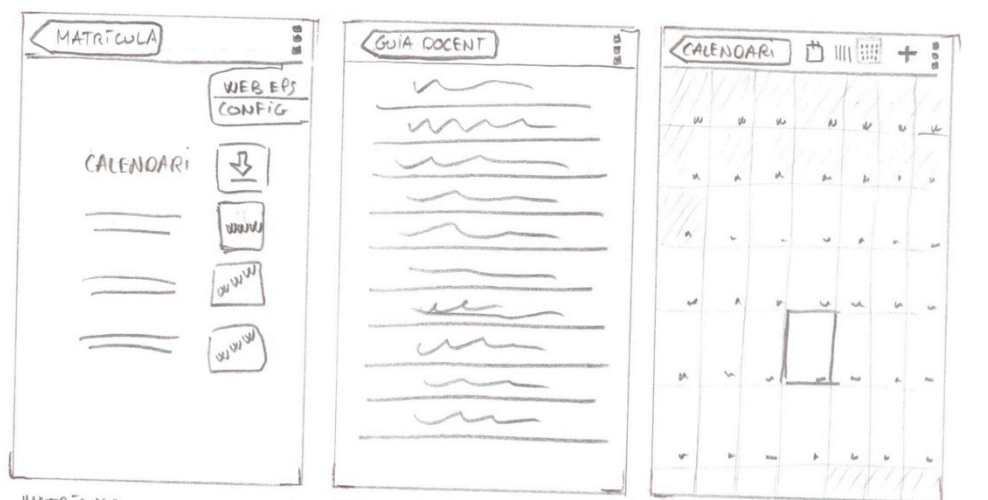
PROTOTIP DE PAPER v2.1

pagina 1



AGENDA
- Consulta d'una activitat de l'agenda.

NOTÍCIES
- Consulta d'una notícia.



MATRÍCULA
- Descàrrega del calendari de matriculació.
- Consulta dels tràmits per fer l'automatrícula.
- Accés a la secció de matrícula de la web.

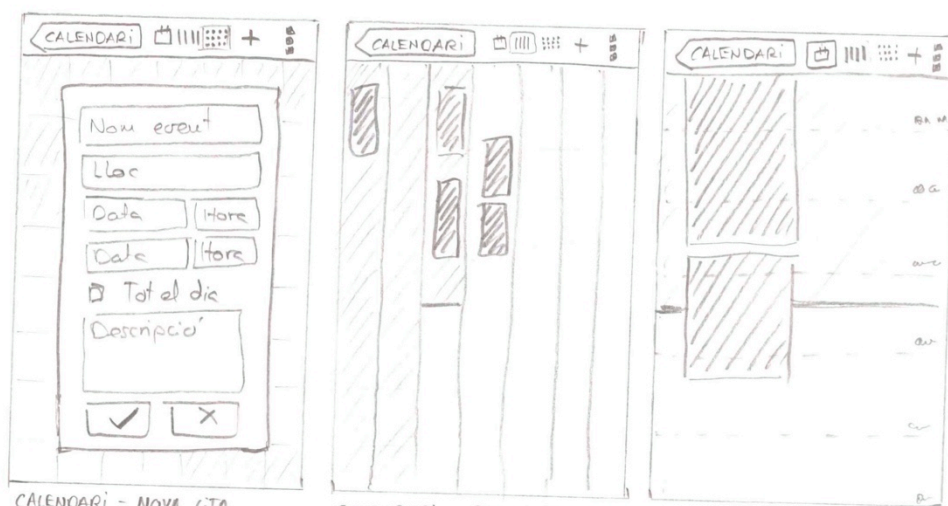
GUIA DOCENT
- Consulta de la guia docent d'una assignatura.
- Consulta d'una guia docent d'un any anterior en la web.

CALENDARI - MES.
- Consulta de les cites del mes.
- Creació d'una cita personal.

Figura 24. Prototip de paper versió 2, pàgina 1

PROTOTIP DE PAPER v2.1

pagina 2

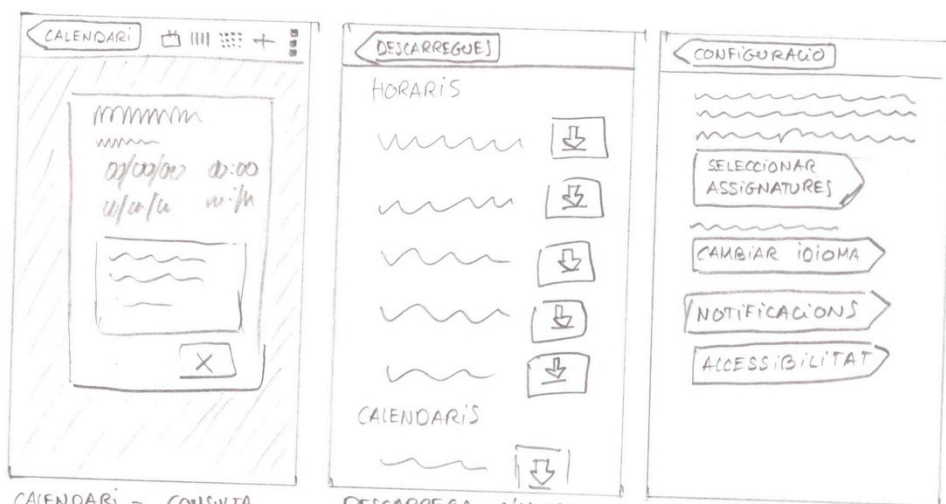


CALENDARI - NOVA CITA.

CALENDARI - SETMANA

CALENDARI - DIA

- Consulta dels detalls/modificacions d'una cita.
- Creació d'una cita personal
- Consulta de les cites de la setmana/dia.



CALENDARI - CONSULTA ACTIVITAT.

DESCARREGA D'HORARIS I CALENDARIS

CONFIGURACIÓ

- Descarrega d'horaris i calendaris.

Figura 25. Prototip de paper versió 2, pàgina 2.

PROTOTIP DE PAPER v2.1

pàgina 3



Figura 26. Prototip de paper versió 2, pàgina 3

Després de la creació del prototip de paper es planteja la realització d'un test amb usuaris. Amb aquest objectiu es crea un prototip digital per donar una primera imatge més real i més propera a la versió definitiva. L'esquema cromàtic i els icones no són del tot definitius però aquesta maqueta ens permet amb antelació saber quins dissenys d'icones o fons de pantalla es necessitaran i les seves proporcions. En la Figura 27 es poden veure alguns dels elements del prototip. La resta es poden consultar en l'Annex III (pàgina 147).

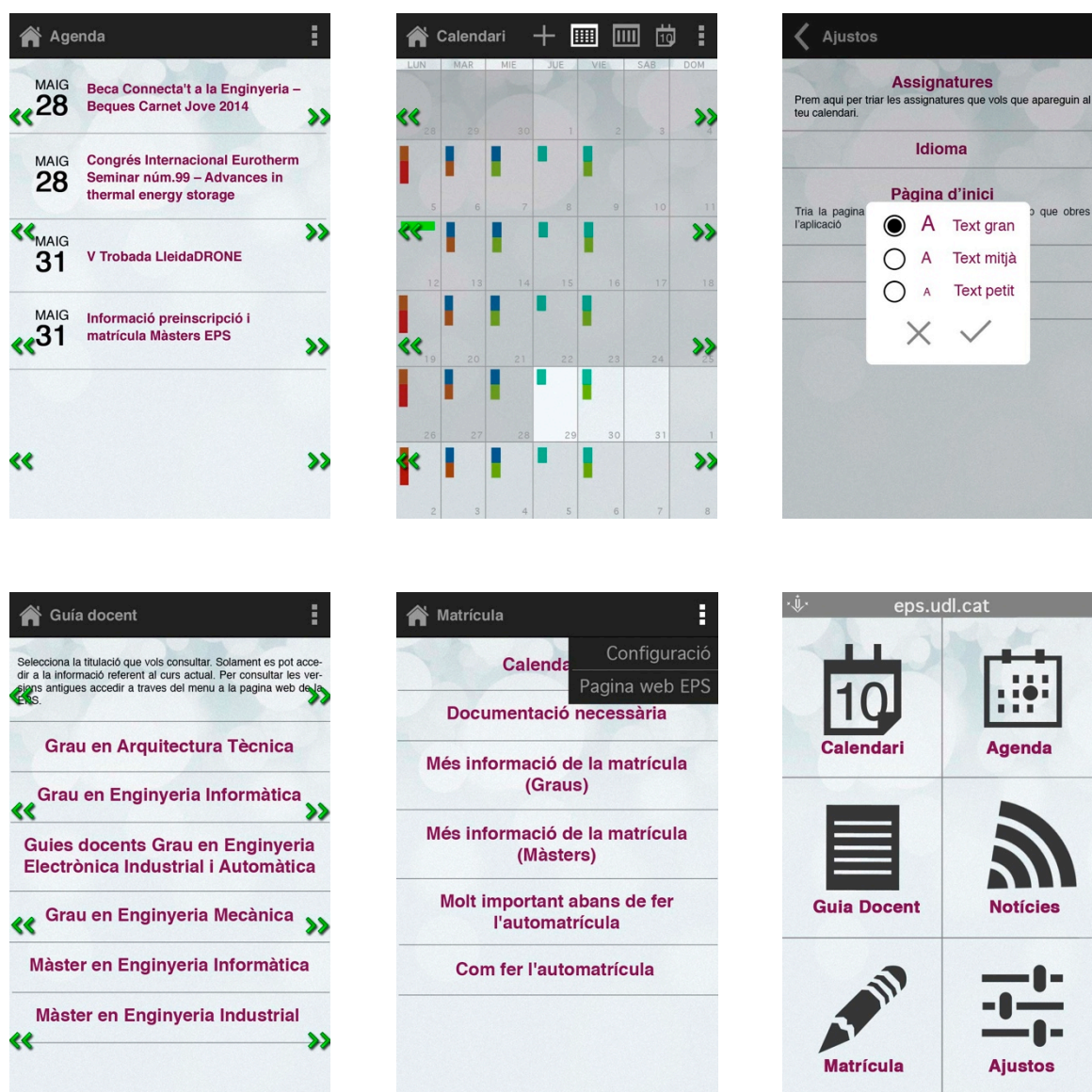


Figura 27. Algunes de les interfícies del prototip digital. Es pot veure la resta en el Annex III (p. 147).

S'han creat dues versions de maqueta digital diferenciades per l'existència o no de la pantalla de Menú. La diferència bàsica, a part que en la versió sense menú no hi és la pantalla de menú, és que en la ActionBar no tenim l'opció d'anar endarrere a la pantalla anterior amb la icona de la casa, sinó que hi ha la icona de l'aplicació i el nom de la secció solament (Figura 28).

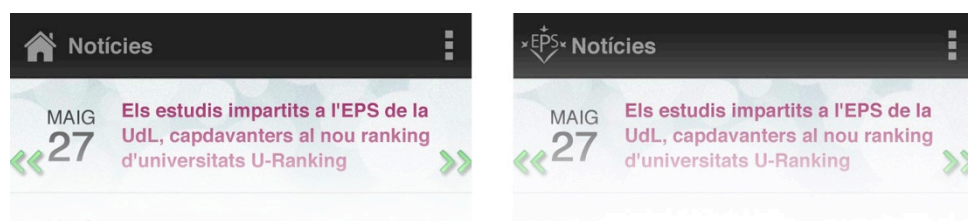
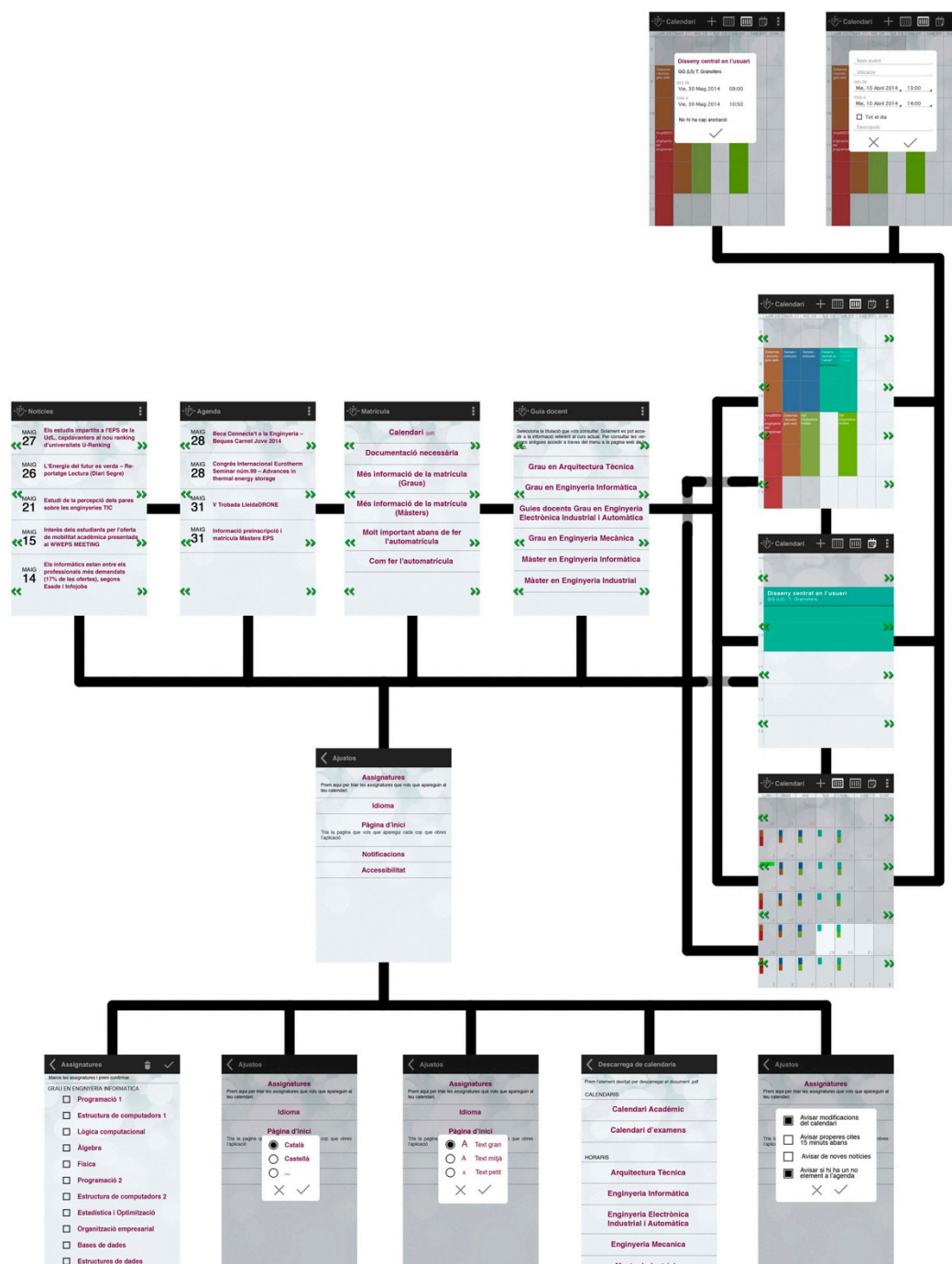


Figura 28. Detall de la barra ActionBar amb la icona de l'aplicació o de inici.

La creació d'aquest prototip digital es realitza amb Photoshop CS5 Extended imitant el tema *Holo Light Dark ActionBar* que Android proporciona com a estàndard.

6.3. Storyboard navegacional¹⁴.

6.3.1. Versió sense menú.

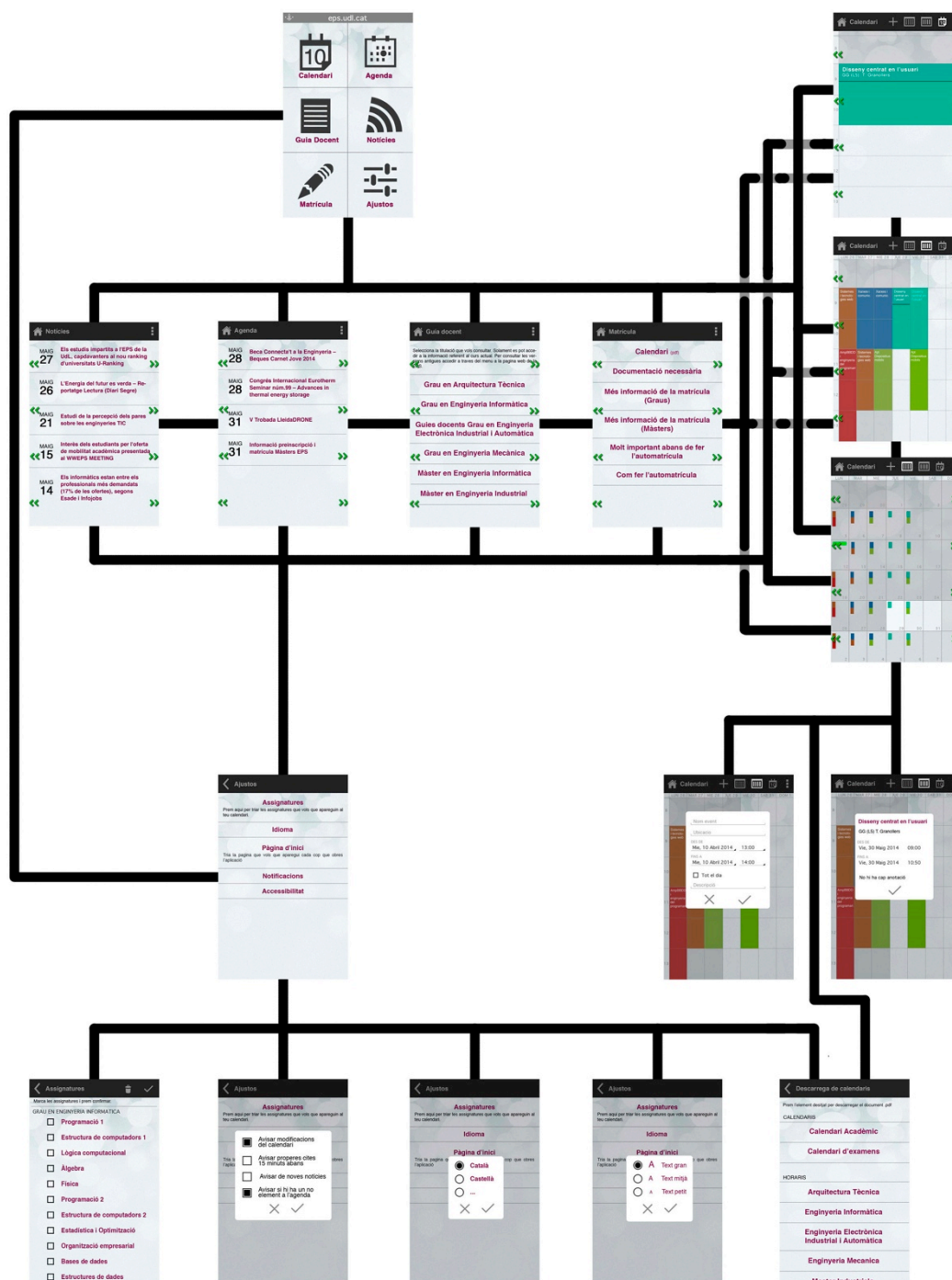


Gràfic 4. Storyboard del prototip digital versió sense menú.

¹⁴ Per més informació:

<http://www.grihotools.udl.cat/mpiuu/mapastoryboard-navegacional-whiteboarding/>

6.3.2. Versió amb menú.



Gràfic 5. Storyboard del prototip digital versió amb menú.

7. Tests amb usuaris.

7.1. Preparació.

Per tal d'avaluar que es compleixin els objectius d'usabilitat es disposa la realització d'un recorregut cognitiu amb usuaris. Principalment s'observarà la facilitat d'aprenentatge dels usuaris i l'agilitat de navegació en l'aplicació. El punt de la càrrega cognitiva ja està controlat al determinar en el disseny l'estructura de les pàgines i, a excepció de l'accés a "configuració", ja es compleix la premissa d'arribar a qualsevol secció amb menys de tres moviments.

7.1.1. Material.

El test es realitzarà directament en un dispositiu mòbil ja que com més real sigui l'entorn més fiable serà el resultat.

Marca:
Motorola

Model:
Moto G

Versió de Android:
Kit Kat 4.4.3
(en el moment del test)

Memòria RAM:
1GB

Memòria interna:
8GB

Processador:
Qualcomm MSM8x26
quad-core A7 1.2GHz
(Adreno 305 450MHz GPU)



Figura 29. Motorola Moto G.

7.1.2. Selecció de software i configuració.

Originalment, la realització d'un test amb un prototip de paper és tan senzill com tenir totes les interfícies per separat i, mitjançant una numeració en tots els botons o amb les indicacions del director del test, quan l'usuari realitza una acció es realitza la resposta del sistema manualment. Per exemple, si estem en un punt que hi ha l'opció de tornar enrere, al prémer en aquell botó es va a la pàgina on era abans de forma manual.

Amb el temps s'han creat diferents aplicacions que permeten la realització del test en el mateix dispositiu com si es tractés de la aplicació real i amb el mateix prototip de paper. Aplicacions que s'han provat com el *POP Prototyping on Paper*¹⁵ o el *Launch*¹⁶ permeten fer una foto directa del paper o importar una imatge de la galeria del dispositiu i des d'aquesta imatge marcar els botons i definir el destí d'aquell botó. En la Figura 30 es veu com es realitza amb l'aplicació Pop, però la versió actual 2.0.

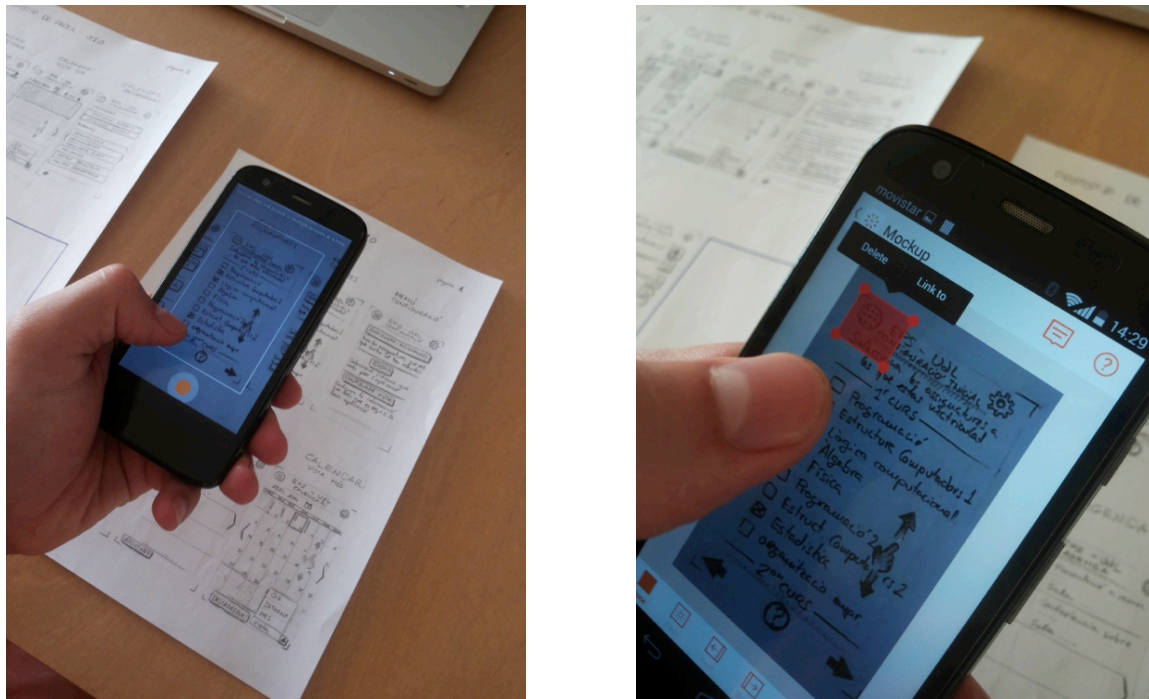


Figura 30. Utilització del programa Pop Prototyping on Paper. A la dreta es veu el moment de fer una foto a una de les interfícies del prototip de paper i a l'esquerra es realitza un enllaç d'un espai determinat de la imatge a una altra pàgina.

¹⁵ <https://popapp.in/>

¹⁶ <https://app.getlaunch.com/>

Inicialment es va intentar la realització del test amb el POP Prototyping on Paper però la configuració de la navegació s'havia de realitzar en el mateix dispositiu mòbil i en el moment de la prova de l'aplicació donava bastants problemes i alguns dels enllaços que havíem configurat no funcionaven. Es va estudiar també la utilització de *Prototyper Pro* de *Justinmind*¹⁷ però era una aplicació molt més complexa perquè està preparat per realitzar prototips digitals quasi funcionals.

Més tard s'ha trobat l'aplicació anomenada *Launch*. A diferència de *POP*, aquesta et permet realitzar la configuració de tota la navegació des de web. La configuració és molt més ràpida i còmoda, ja que des de la web visualment és molt més entenedor, i en el test funciona perfectament. Totes dues aplicacions es sincronitzen en qualsevol dispositiu per tant, es podria fer el test en qualsevol dispositiu i ofereixen versions gratuïtes que permeten la realització del test sense problemes.

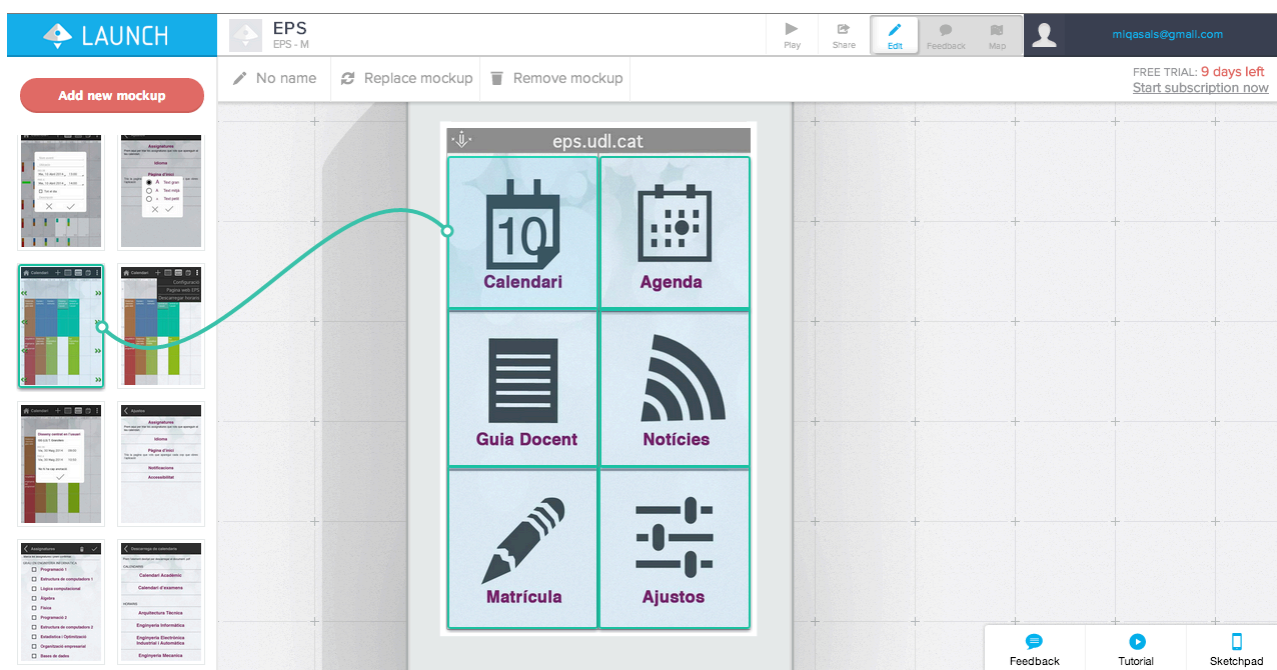


Figura 31. Captura de pantalla de l'aplicació Launch amb gestió a través del navegador web.

¹⁷ www.justinmind.com/prototyper/widgets-mobile

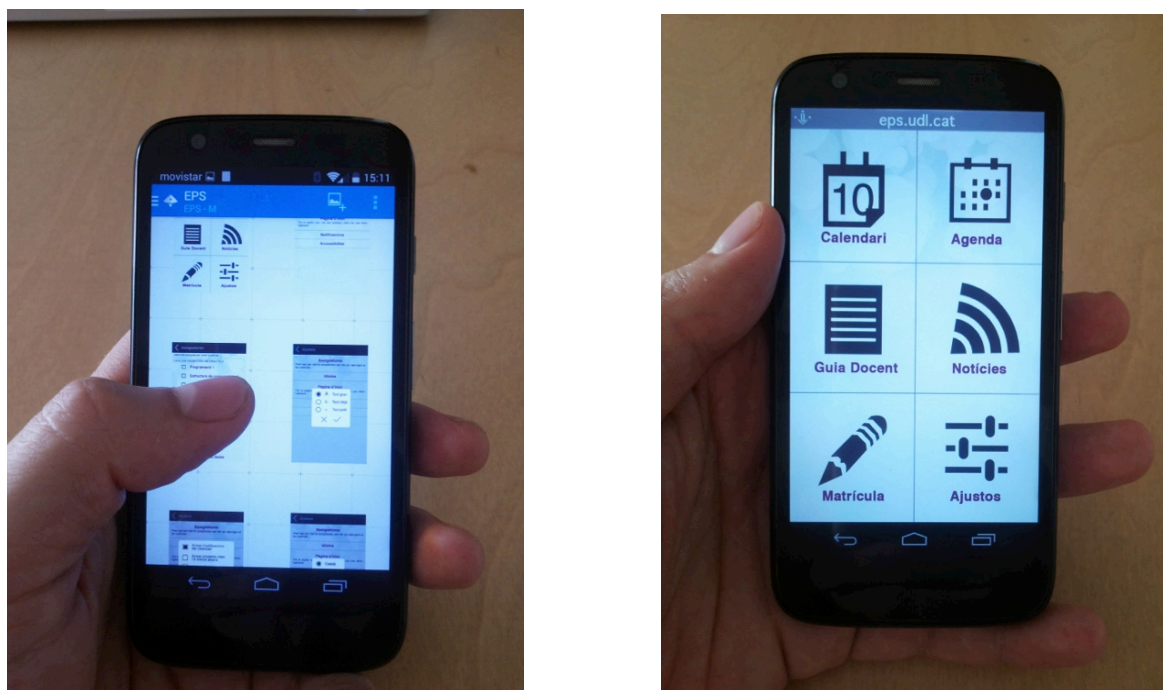


Figura 32. Utilització de l'aplicació Launch a través del dispositiu mòbil. A l'esquerra es la vista que ofereix l'aplicació i a la dreta es veu la pantalla en el moment de realització del test.

7.1.3. Selecció de la mostra.

El test es realitza amb un grup de catorze alumnes de 3er del Grau d'Informàtica de la Universitat de Lleida. S'ha escollit aquest grup perquè és un perfil clar de futur usuari de l'aplicació, pels alts coneixements en la matèria d'Interacció Persona - Ordinador i Disseny Centrat en l'Usuari i amb la possibilitat que aportin observacions e idees en el tema.

La mostra es divideix en dos grups que realitzaran el test amb les dues versions diferents del prototip. Un grup realitzarà el test amb la versió que disposa de pantalla inicial de Menú i el segon grup ho realitzarà amb la versió que no disposa d'aquesta pantalla. S'aprofitarà una classe d'aquests alumnes per realitzar el test en una taula separada d'un en un sense que l'usuari sàpiga que hi ha dues versions diferents.

Per millorar la representativitat de la mostra s'hauria de realitzar el test amb els diferents usuaris que s'han definit a l'anàlisi de requisits i de diferents nivells de coneixements però no es disposa de suficient temps per crear tal grup i realitzar els tests.

7.1.4. Plantilla del test.

Per la realització del text es planteja a cada usuari cinc tasques que ha de realitzar de forma individual i sense saber res de l'aplicació. Quan completi cada tasca ha de valorar en un qüestionari la dificultat que li ha suposat l'assoliment d'aquesta valorant amb 0 la tasca com a "molt difícil" i 5 com a "molt fàcil". A la part final del qüestionari es consulta a tots dos grups si considera necessària la pantalla de menú, indicant que en l'aplicació finalitzada es tindrà l'opció de configurar la pantalla d'inici que es vulgui. Es deixarà un apartat de comentaris on puguin anotar qualsevol idea o detall que creguin convenient.

En la següent Figura 33 es mostra la plantilla que omplien els usuaris.

TFC: Elaboració d'una aplicació Android de la web de l'EPS seguint el procés de Disseny Centrat en l'Usuari MPLu+a.

Test d'usuari 29/05/2014

Encercla les teves respostes.

1- Vols anar a la V Trobada LleidaDRONE, per tant, busca el dia, l'hora i el lloc.

Puntua el grau de dificultat que t'ha suposat aquesta tasca:

(Molt difícil) 0 1 2 3 4 5 (Molt fàcil)

2- Comprava si hi ha alguna anotació nova a la propera assignatura que tens avui.

Puntua el grau de dificultat que t'ha suposat aquesta tasca:

(Molt difícil) 0 1 2 3 4 5 (Molt fàcil)

3- Configura les assignatures que vols que surtin al calendari del dispositiu i que vols rebre notificacions.

Puntua el grau de dificultat que t'ha suposat aquesta tasca:

(Molt difícil) 0 1 2 3 4 5 (Molt fàcil)

4- Consulta el pla formatiu de la teva titulació per saber quantes assignatures et queden per completar-la.

Puntua el grau de dificultat que t'ha suposat aquesta tasca:

(Molt difícil) 0 1 2 3 4 5 (Molt fàcil)

5- Has d'anar a fer la matriculació de l'any vinent. Que necessites?

Puntua el grau de dificultat que t'ha suposat aquesta tasca:

(Molt difícil) 0 1 2 3 4 5 (Molt fàcil)

Comentaris.

En l'aplicació es pot configurar la pantalla d'inici que vols que surti en el moment d'iniciar. Creus que es necessari que hi hagi una pantalla "Menú" on es pugui accedir directament a qualsevol de les seccions?

SI

NO

Figura 33. Plantilla de qüestionari del test amb usuaris.

7.1.5. Recorregut amb prototip digital amb Menú.

Punt 1. Obtenir informació sobre “V Trobada LleidaDRONE”.

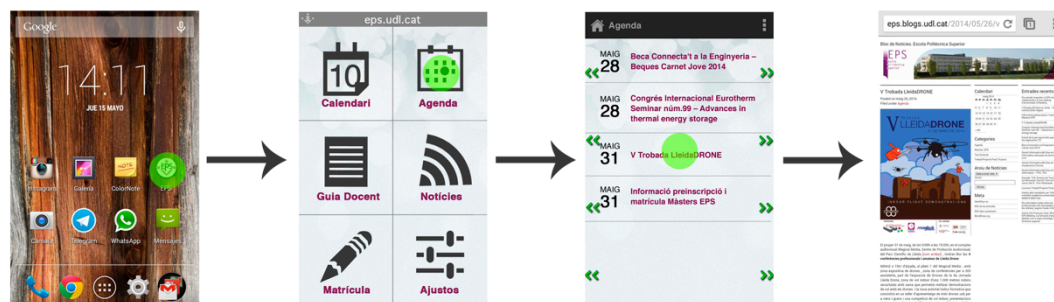


Figura 34. Recorregut per a la qüestió 1 del test amb versió amb menú.

Punt 2. Comprova si hi ha alguna anotació nova a la propera assignatura que tens.

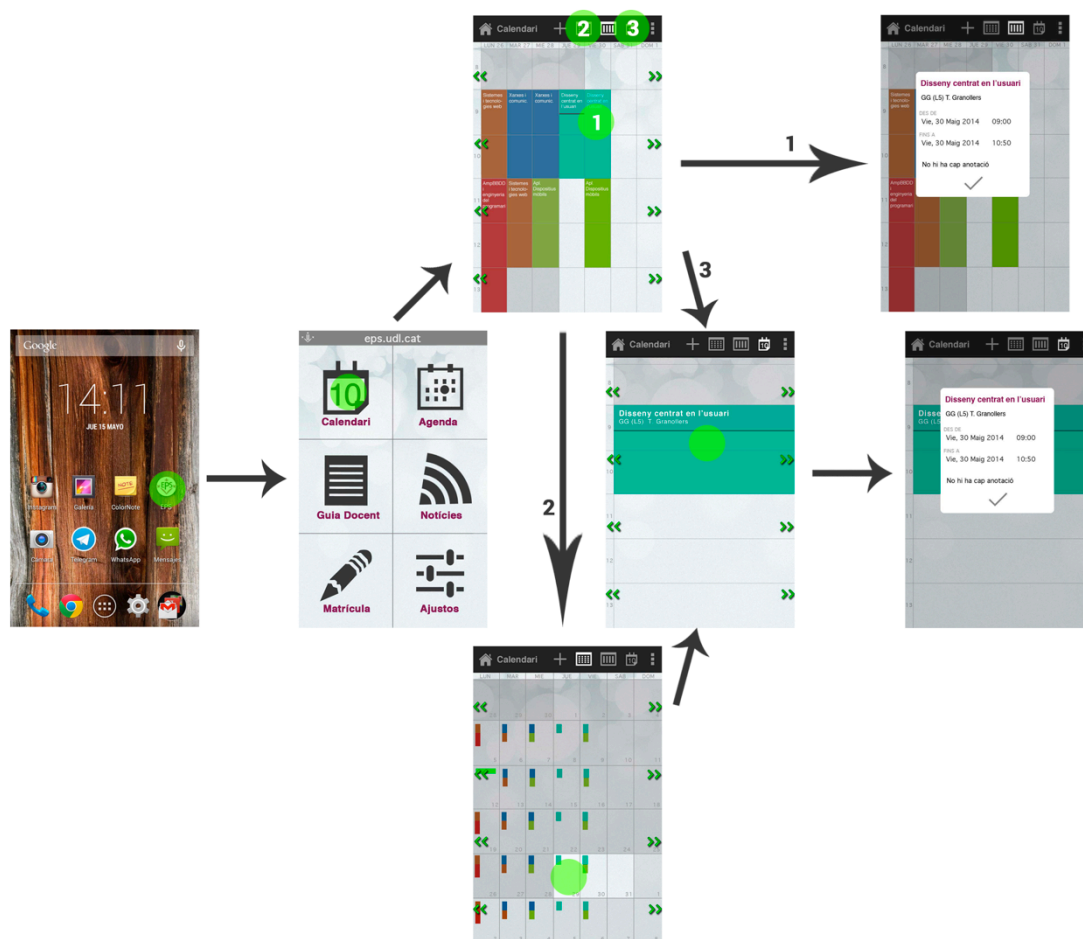


Figura 35. Recorregut per a la qüestió 2 del test amb versió amb menú.

Punt 3. Configura les assignatures que vols que surtin al calendari del dispositiu.

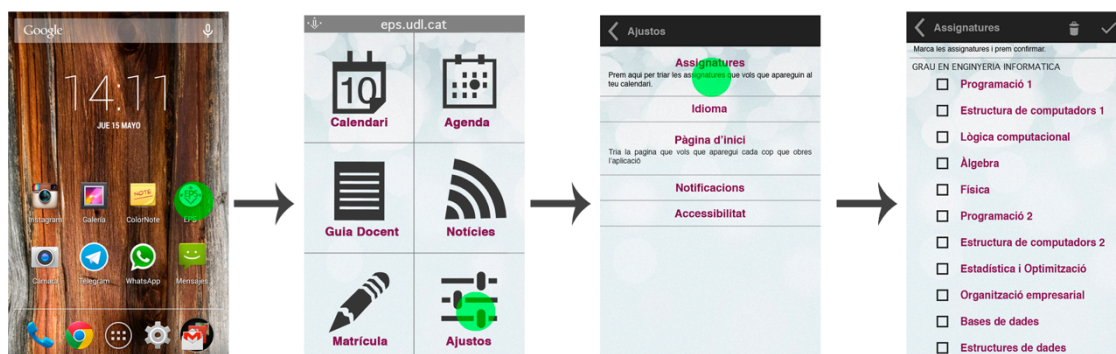


Figura 36. Recorregut per a la qüestió 3 del test amb versió amb menú.

Punt 4. Consulta el pla formatiu de la teva titulació per saber quines assignatures queden per completar-la.

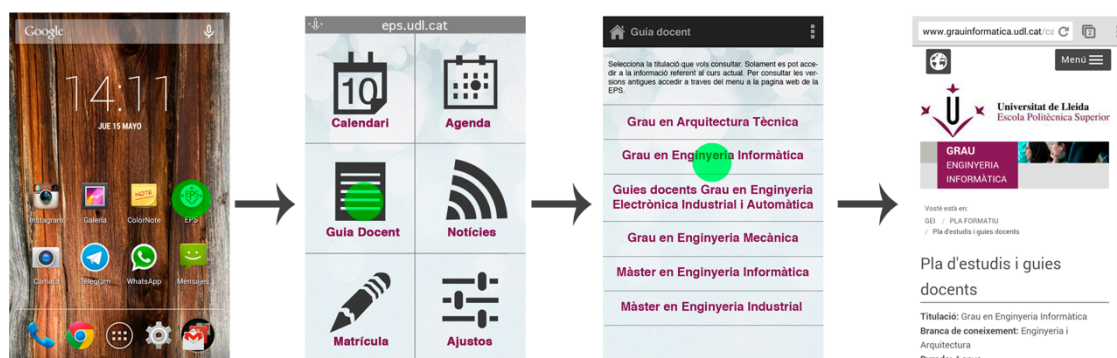


Figura 37. Recorregut per a la qüestió 4 del test amb versió amb menú.

Punt 5. Has d'anar a fer la matriculació de l'any vinent. Que necessites?

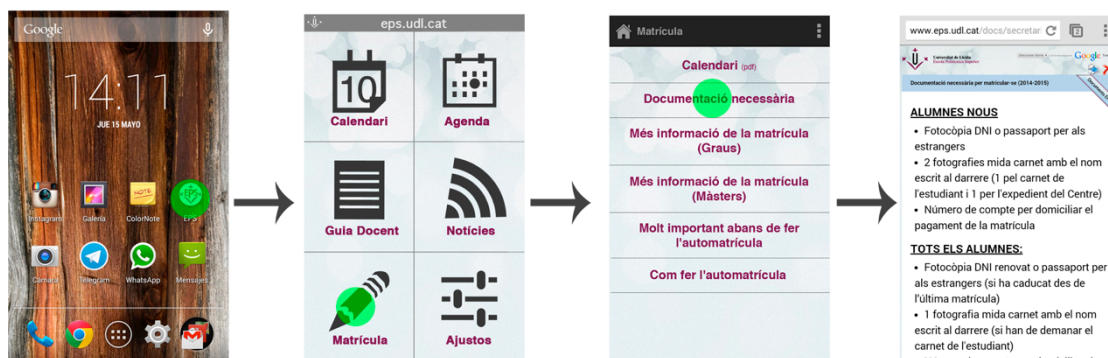


Figura 38. Recorregut per a la qüestió 5 del test amb versió amb menú.

7.1.6. Recorregut amb prototip digital sense Menú.

Punt 1. Obtenir informació sobre “V Trobada LleidaDRONE”.



Figura 39. Recorregut per a la qüestió 1 del test amb versió sense menú.

Punt 2. Comprova si hi ha alguna anotació nova a la propera assignatura que tens.



Figura 41. Recorregut de la qüestió 2 del test amb versió sense menú.

Punt 3. Configura les assignatures que vols que surtin al calendari del dispositiu.

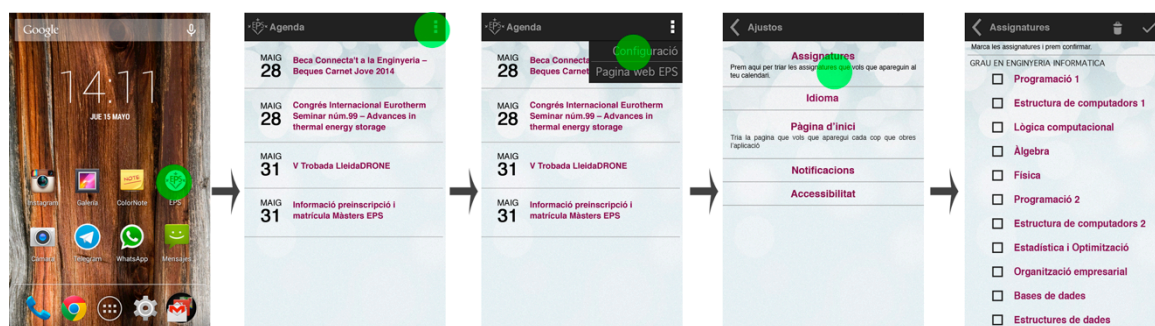


Figura 40. Recorregut per a la qüestió 3 del test amb versió sense menú

Punt 4. Consulta el pla formatiu de la teva titulació per saber quines assignatures queden per completar-la.

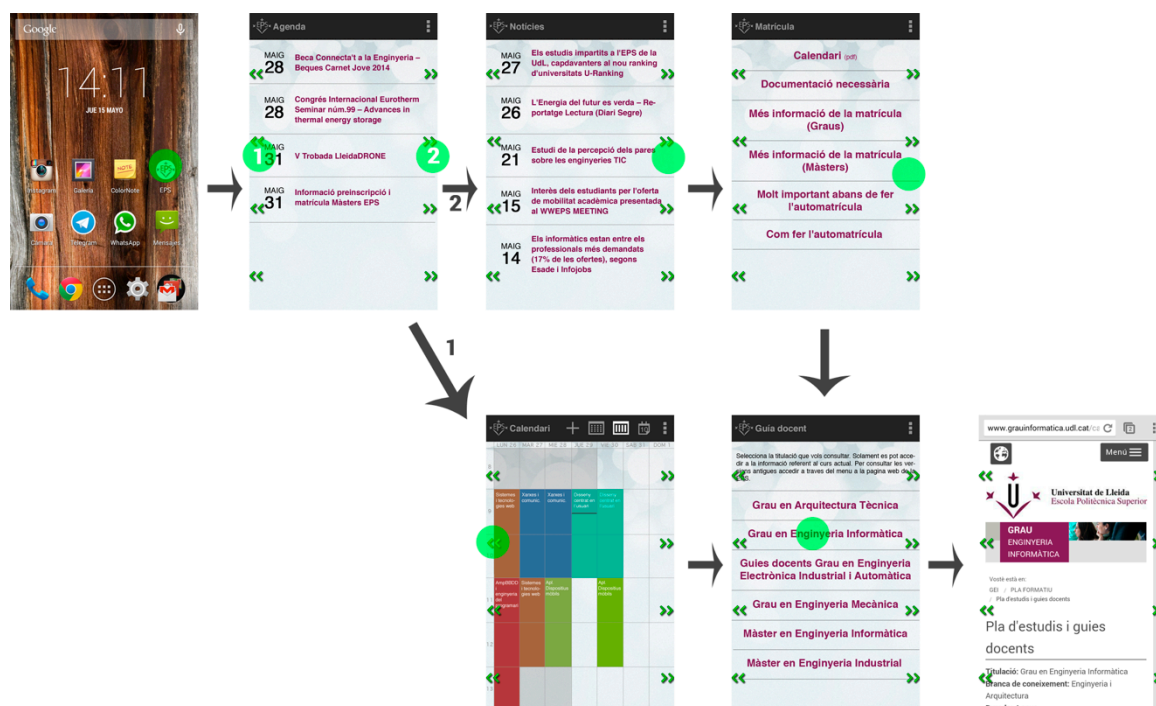


Figura 42. Recorregut per la qüestió 4 del test amb versió sense menú.

Punt 5. Has d'anar a fer la matriculació de l'any vinent. Que necessites?

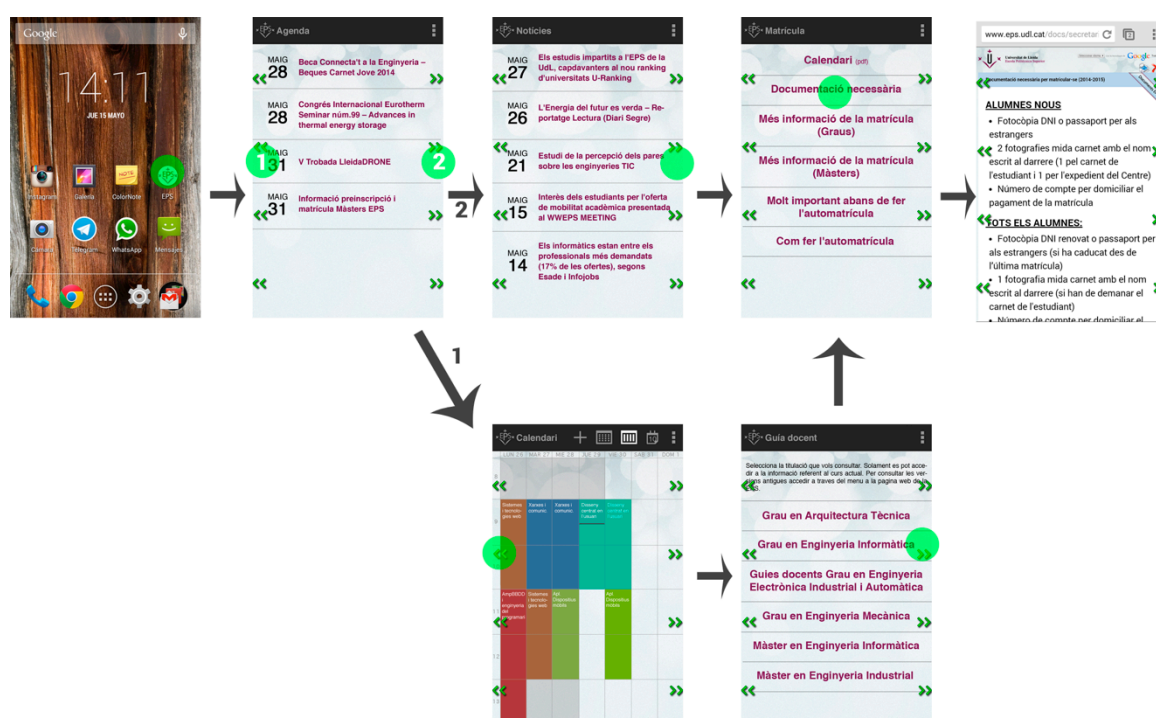
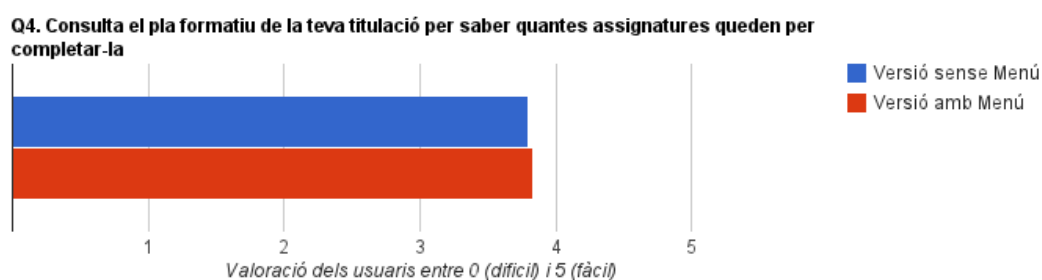
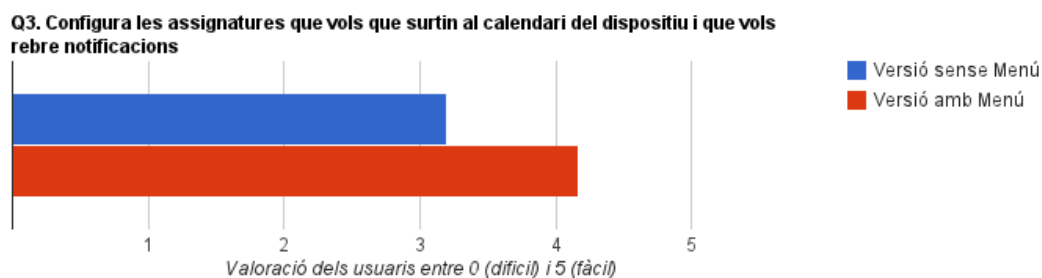
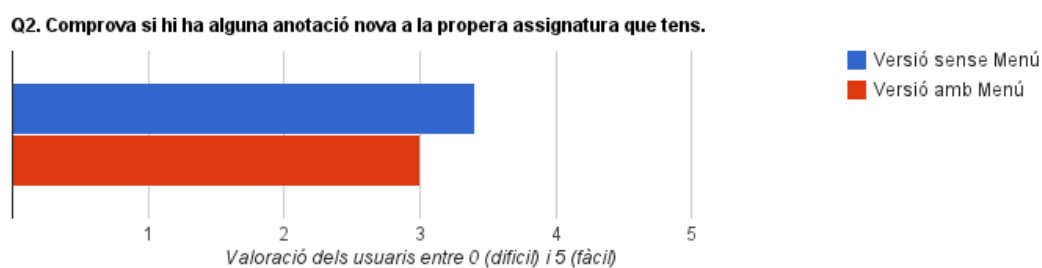
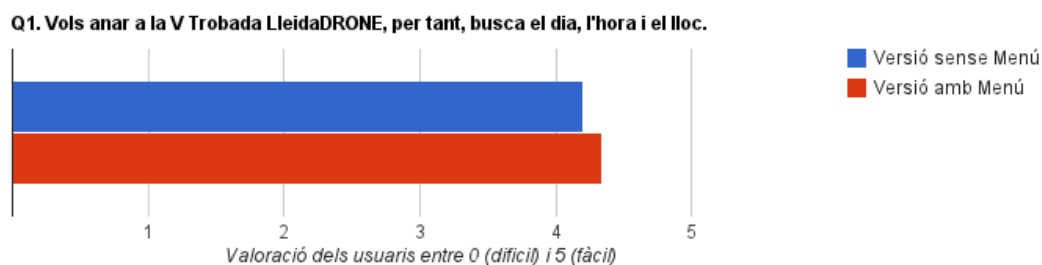


Figura 43. Recorregut per la qüestió 5 del test amb versió sense menú.

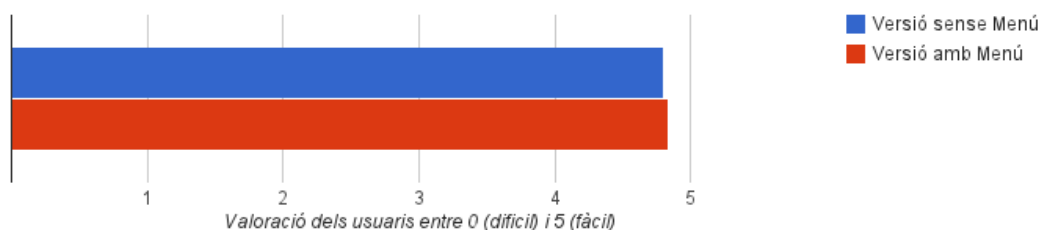
7.2. Resultats.

7.2.1. Valoracions dels usuaris en el test.

Realitzat el test s'han extret les següents valoracions:

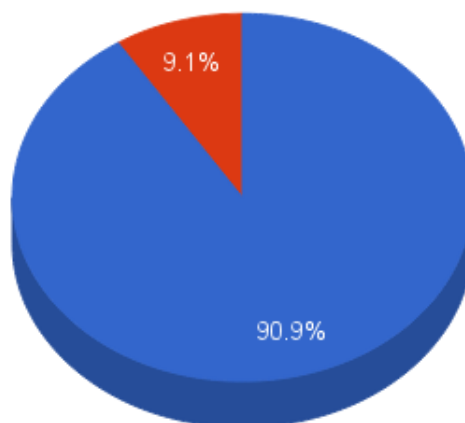


Q5. Has d'anar a fer la matriculació de l'any vinent. Que necessites?



En l'aplicació es pot configurar la pantalla d'inici que vols que surti en el moment d'iniciar. Creus que es necessari que hi hagi una pantalla "Menú" on es pugui accedir directament a qualsevol de les seccions?

■ si ■ no



Comentaris en la versió sense menú.

- Quan entres a l'aplicació no tens cap indicació d'on estàs situat.
- Trobo a faltar que indiqui el lloc on estàs situat. (També comenta l'usuari que si hi hagués una marca de posició, ja sigui pestanyes o algun altre tipus d'indicació, no seria necessària la pantalla de Menú.)

Comentaris en la versió amb menú.

- L'apartat 2, al calendari no es veu clar que prement en les assignatures et sortirà un quadre de text anotació, falta alguna icona.
- Indicats per l'usuari: El fet que sigui un prototip i la majoria de funcionalitats no estan operatives provoca una mica de confusió al realitzar el test.

7.2.2. Anàlisi dels resultats.

Un cop realitzat el test s'ha decidit que la versió amb menú és millor ja que tots els usuaris excepte un han considerat necessària la pàgina menú. Un dels usuaris que ha realitzat el test amb la versió sense menú, ha proposat que un indicador indiqui on ets en cada moment dins de l'aplicació en lloc del menú però, donat que àmpliament es considera que és millor amb menú, s'hi inclourà. Igualment hi haurà la possibilitat que l'usuari configuri la pàgina amb que vol iniciar l'aplicació.



Figura 44. Vista de la interfície menú del prototip digital.

D'acord amb la valoració dels usuaris la mitjana obtinguda és 3,96 sobre 5, per tant la navegació i l'aprenentatge és considerada senzilla i fàcil.

Hi ha tres qüestions que estan una mica per sota de 4. És degut a que, com alguns usuaris van observar, el test en una maqueta digital on dona la sensació que estàs provant l'aplicació real és, en ocasions, confús i dona lloc a errors en la navegació. La qüestió 2, que és comprovar l'anotació en la següent cita, n'és un exemple clar. La interacció amb la secció del calendari està pensada i dissenyada seguint les pautes del calendari del sistema Android. Inclús, en d'altres sistemes operatius l'esquema és molt similar, tant en aparença visual com en interacció. Tot i això, als usuaris els ha costat veure que prement sobre la següent cita s'obre una finestra amb la informació d'aquesta cita. El fet que la imatge del prototip fos fixa i solament obria la finestra de la cita indicada com a següent, que era l'endemà, duia a la confusió.

Aquí s'exposen alguns exemples de calendaris de diferents Sistemes Operatius de dispositius mòbils (Figura 45 – 49).



Figura 45. Calendari Windows Phone 8.

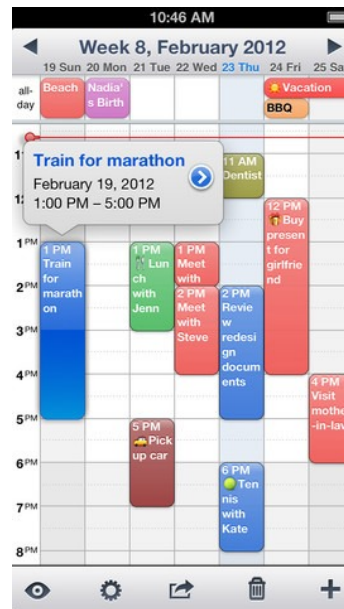


Figura 46. Calendari Android

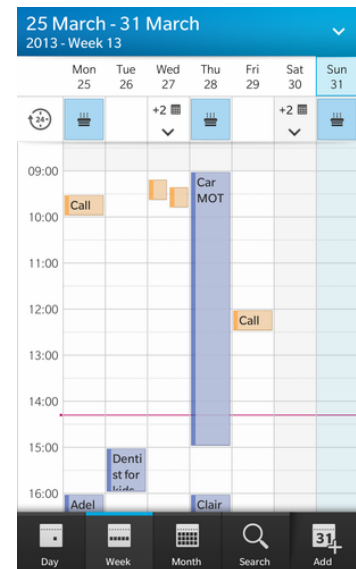


Figura 47. Calendari Blackberry Z10.

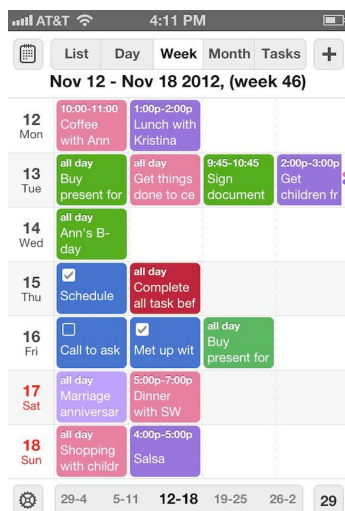


Figura 48. Calendari iOS7 (Apple).



Figura 49. Calendari del Prototip Digital utilitzat en el test.

Com es pot observar, pràcticament tots es basen en el mateix esquema. Sí que en la versió del prototip pot faltar algun tipus de marc o separador entre activitats i que doni la sensació que cada activitat és un objecte diferent però es considera que el disseny és el correcte i en la versió real aquest problema no hi serà.

També en el punt 3 del qüestionari, més d'un usuari ha indicat que no sabien exactament què fer quan eren en l'apartat de configuració de les assignatures perquè no podien marcar cap d'elles. És part del mateix problema que el cas del calendari, qüestió 2. Però, no obstant, en el moment que un usuari s'instal·la l'aplicació i entra per primer cop seria bo que aparegués una indicació d'ajuda apuntant els aspectes que es recomanable que l'usuari configuri o que directament un assistent actués com a guia. Aquest assistent ha de poder ser seguit en el moment que es vol, suposant que potser l'usuari no vol configurar-ho la primera vegada que entra, o inclús poder desactivar l'assistent.

7.2.3. Conclusions del test.

- Versió definitiva amb el menú d'inici i accessible des de cada secció.
- Recomanable assistent de configuració en el moment que s'instal·la l'aplicació.
- Navegació per l'aplicació i assoliment d'objectius valorat positivament.

8. Programació.

8.1. Gestió del projecte.

Els coneixements de programació en Android són molt limitats, ja que mai havia realitzat cap tipus de formació sobre el tema. Amb l'ajuda de bibliografia especialitzada adquirida personalment i Internet es realitza la codificació el més semblant possible als prototips que es tenia pensats inicialment.

En un principi es deixa de banda la programació en fragments. Des de la versió d' Android 3.0 API 11 existeix la possibilitat de crear interfícies de forma que es pot dividir la pantalla i modificar únicament la part que es vol en el moment d'interactuar. Això permet reduir el consum de recursos, ja que sols es modifica part de la pantalla però és bastant més costos en programació. És una tècnica més enfocada a tauletes, ja que dividir la pantalla en un telèfon, excepte en aparells de gamma alta amb pantalles de grans dimensions, no ens aporta cap benefici.

Primer es realitza la programació en XML de totes les interfícies i es crea l'estructura de totes les seccions de l'aplicació. Després s'anirà realitzant la programació en Java secció a secció. Algunes seccions tindran una estructura similar entre elles per tant, s'aprofita el codi modificant les referències a la informació que es mostra i a la plantilla XML que el compon. En l'apartat de desenvolupament (8.2) s'explica, agrupant les seccions amb igual estructura, com s'ha realitzat la programació i les possibles dificultats i canvis que s'han hagut de realitzar.

Tota la programació es realitza amb l'estil *Holo Light Dark ActionBar* i quan ja es té tota la programació realitzada es modifica l'estil per personalitzar-lo. El disseny de la icona de l'aplicació i les dels botons del menú es realitza quan ja es té finalitzada la programació de l'aplicació.

8.2. Desenvolupament

8.2.1. Menú inicial

Estructura.

En el prototip, el menú inicial estava format per un botó a la part superior, d'accés a la pàgina web, i d'una graella de 3 x 2 botons amb totes les seccions més la de preferències. Per crear aquesta estructura de botons hi ha diferents opcions bàsiques:

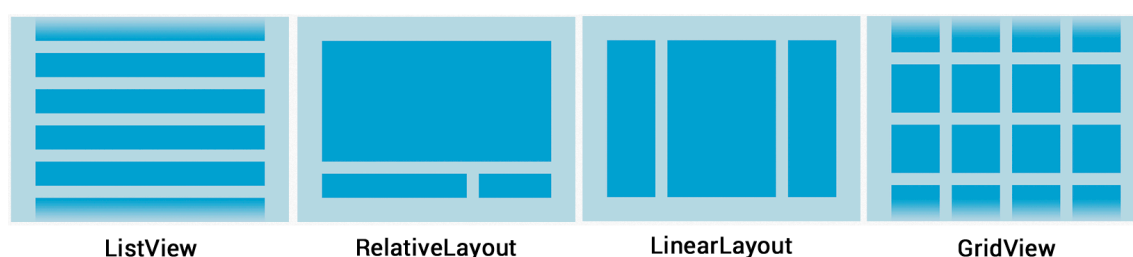


Figura 50. Estructures més bàsiques utilitzades en Android.

Inicialment es realitza amb la plantilla *GridView* però no es disposa de la suficient informació per ocupar tot l'ample de la pantalla. Els botons ocupen únicament l'espai del seu contingut i si es configuren amb una mida fixa no s'adapten a les diferents pantalles. Aquest problema també es tenia amb el *RelativeLayout*. Per tant, es va optar per una construcció de *LinearLayouts* amb l'atribut *android:layout_weight* que permet repartir dinàmicament la mida dels elements distribuint el l'espai segons el "pes" que li has indicat. Aquest atribut, en *GridView* o *RelativeLayout* no es podia aplicar.

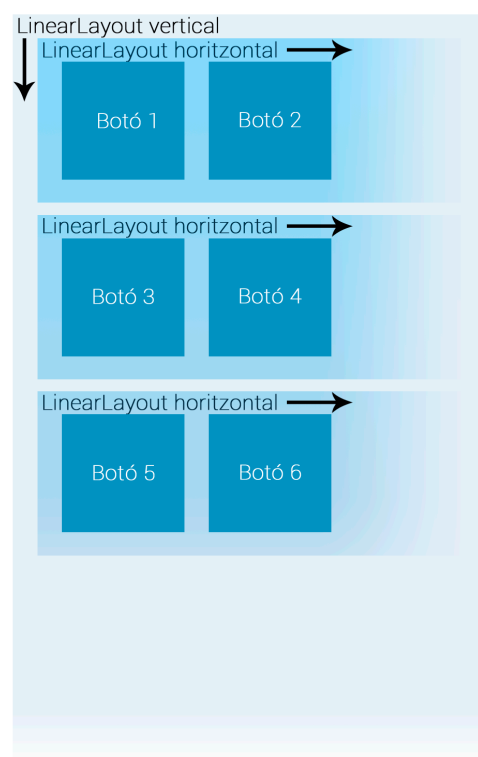


Figura 51. Estructura XML per a la interfície del menú.

Dificultats i modificacions.

Inicialment es volia dissenyar amb pantalla completa sense *ActionBar* però suposa dos problemes: l'emblema de l'aplicació i la transició entre interfícies. L'emblema de l'aplicació no seria visible enlloc i el fet que hi sigui ajuda a identificar l'aplicació. El problema en la transició és què en el moment que es toca un botó per canviar a una altra secció es passa d'una vista sense *ActionBar* a una altra que sí que en té. Com a resultat, l'aplicació primer carregava a pantalla completa la nova interfície i després l'encongia per poder incloure la barra, apareixent aquesta per últim. Al tornar enrere passava a la inversa, primer carregava la pàgina del menú i després ampliava a la pantalla completa. Tot plegat dona sensació d'inestabilitat i semblava que s'estigui produint algun tipus d'error. Per tant, finalment, s'opta per crear el menú amb *ActionBar*.

El fet que es deixés visible l'*ActionBar* dona l'opció que el botó d'accés a la pàgina web de l'EPS passi a ser una icona d'aquesta o al menú *Overflow* per alliberar espai (veure la secció d' *ActionBar*, pàgina 107).

Un altre punt que ha comportat dubtes és el tema de la pantalla giratòria. En el moment que es feia el disseny amb pantalla complerta es volia mantenir en la mateixa posició tots els botons, ja que si gires el dispositiu l'estructura es manté però amb les mides totalment diferents de forma que els botons adapten les mides al nou espai. Si es crea l'estructura en horitzontal per tal que mantingues les mides duplicant el document *menu.xml*, però aquest cop en la carpeta *res>layout-land>menu.xml*, es continua reproduint l'animació en el moment de fer el gir. Finalment, amb la inclusió de l'*ActionBar* i la supressió del botó d'accés web es decideix bloquejar la rotació i si es disposa de temps es crearà l'animació de rotació únicament per la icona i el text dels botons.

Quan ja es tenia programada la interfície es realitza el canvi d'estil i s'inclou les icones als botons. Això provoca que els botons no quedin centrats tal com es desitja i es busca una solució més elegant. En lloc d'utilitzar l'atribut XML *android:layout_weight* es modifica dinàmicament en el moment que es crea l'*Activity*. Calculem les mides disponibles de pantalla descomptant l'espai que ocupa la *statusBar* i l'*ActionBar* amb el següent codi:

```
Display display = getWindowManager().getDefaultDisplay();
Point size = new Point();
display.getSize(size);
int width = size.x;
int ab_dp = dpToPx(48);
int height = (size.y)-ab_dp-getStatusBarHeight();
```

Amb l'alçada i amplada de la pantalla es realitza la modificació de les variables *height* i *width* per dividir la pantalla entre els 6 botons realitzant el següent codi amb cadascun.

```
Button button = (Button) findViewById(R.id.nom_del_boto);
button.setHeight((height/3));
button.setWidth(width/2);
```

La mida de la barra d'estat en píxels s'obté amb la següent funció¹⁸.

```
public int getStatusBarHeight() {
    int result = 0;
    int resourceId=getResources().getIdentifier("status_bar_height"
                                                ,"dimen","android");
    if (resourceId > 0) {
        result = getResources().getDimensionPixelSize(resourceId);
    }
    return result;
}
```

La mida de l'*ActionBar* és 48 *dp* però s'ha de transformar a píxels ja que la funció per obtenir les dimensions de pantalla ens respon amb aquesta unitat. Per fer-ho transformen amb la funció *dpToPx*¹⁹.

```
public int dpToPx(int dp) {
    DisplayMetrics displayMetrics = getBaseContext().getResources()
                                                .getDisplayMetrics();
    int px = Math.round(dp * (displayMetrics.xdpi /
                              DisplayMetrics.DENSITY_DEFAULT));
    return px;
}
```

¹⁸ <http://stackoverflow.com/questions/3355367/height-of-statusbar/3356263#3356263>

¹⁹ <http://stackoverflow.com/questions/8309354/formula-px-to-dp-dp-to-px-android/17410076#17410076>

8.2.2. Matrícula i Guia Docent.

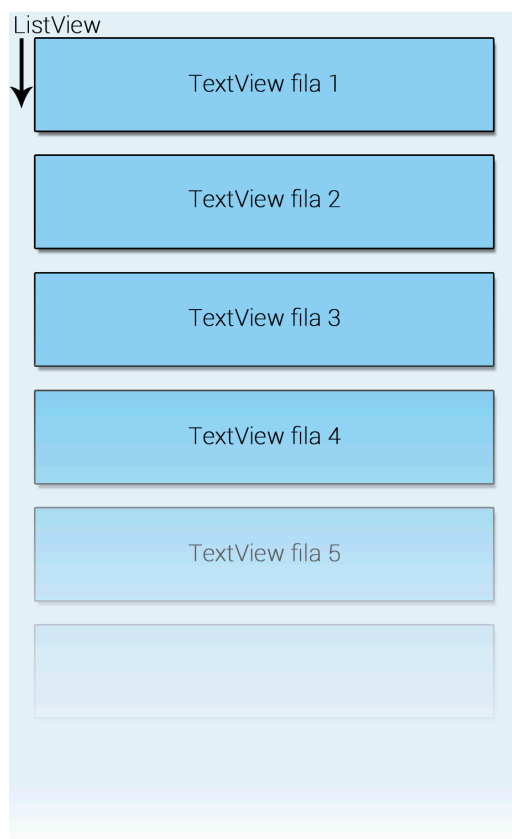


Figura 52. Estructura XML per a les interfícies de Matrícula i Guia Docent.

Estructura.

En les seccions de matrícula i guia docent la decisió ha estat senzilla ja que són dos interfícies molt clares com a llista. Tenim uns elements fixes que són enllaços a diferents direccions on es troba la informació detallada. S'implementa com a estructura un *ListView* senzill on cada fila sol inclou el nom de l'enllaç amb un *TextView* amb les característiques del text que volem mostrar (mides, color...).

Programació.

Es crea una llista simple mitjançant una classe que hereta *ListActivity*. Únicament li passem una taula amb tots els elements de la llista i es mostra mitjançant un *simpleListAdapter*.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    getActionBar().setDisplayHomeAsUpEnabled(true);
    setContentView(R.layout.guia);
    String[] llista = getResources().getStringArray(R.array.GUIA);
    setListAdapter(new ArrayAdapter<String>(this, R.layout.fila,
                                            llista));
}
```

La gestió dels clics s'ha realitzat amb la funció `onListItemClick()`. Es captura el text que conté l'element tocat mitjançant `getListAdapter().getItem()` i diferenciem amb un condicional el text per crear l'*Intent* que obrirà el navegador amb la direcció web corresponent.

```
public void onListItemClick(ListView l, View v, int position, long id) {
    String item = (String) getListAdapter().getItem(position);
    if ( item.contains("TEXT DE L'ELEMENT SELECCIONAT")){
        Intent intent = new Intent(Intent.ACTION_VIEW,
                                   Uri.parse("DIRECCIÓ WEB"));
        startActivity(intent);
    } else if (...) {
        ...
    } ...}
}
```

8.2.3. Agenda i Notícies.

Obtenció de la informació.

Aquesta secció ha estat, deixant de banda calendari, la més complexa de realitzar. S'han de capturar els elements que es mostren a la pàgina web de l'EPS i mostrar-los de forma automàtica i actualitzant-se quan sigui necessari. Aquests elements provenen d'una font RSS del blog de l'EPS (<http://eps.blogs.udl.cat/feed/atom>) que s'obté revisant el codi font de la pàgina web.



Figura 53. Captura de la pàgina web de l'EPS on es veu l'estructura de les notícies.

Teníem dues opcions, realitzar la descàrrega dels esdeveniments cada cop que accedim a la secció o que es guardi el contingut de la font RSS en un arxiu XML. El problema de realitzar la descàrrega cada cop que accedeixes és el consum de dades i es temps de descàrrega i que, en cas que no tinguís connexió, no es carregaria la secció. Que la secció no es pugui carregar per falta de connexió no és problemàtic ja que mostrem els titulars i al seleccionar qualsevol element ens

obre l'enllaç, cosa que tampoc seria possible. El tema del consum de dades sí que és més greu, per tant, s'implementarà la segona opció, la descàrrega del document.

Estructura.

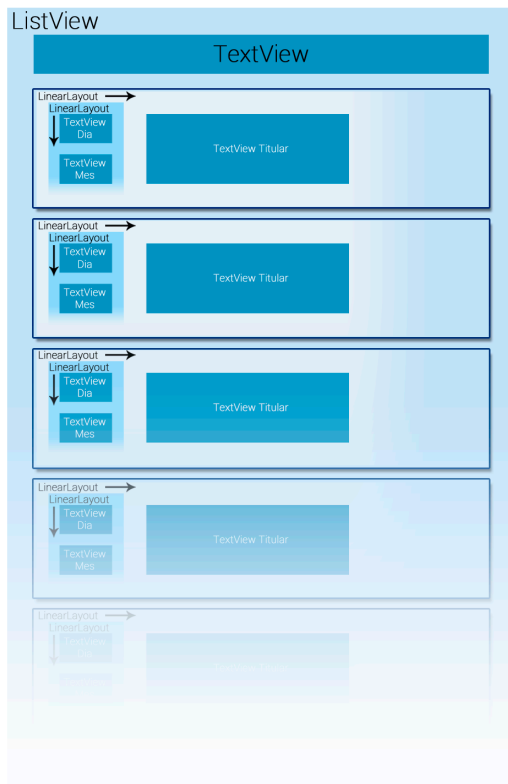


Figura 54. Estructura XML de les interfícies de Agenda i Notícies.

La interfície de la secció és amb forma de llista ja que és un nombre indeterminat d'elements amb les mateixes propietats similar a les seccions de matrícula i guia docent. A diferència d'aquestes no es tracta d'un sol element *ListView*, ja que es vol introduir un *TextView* a la part superior de la llista que mostrarà la data d'actualització i, en cas que no s'hagi descarregat encara el contingut o hi hagi algun tipus d'error, ho s'indicarà aquí. Per poder imitar l'esquema de visualització de la pàgina de l'EPS és necessària la creació d'una llista personalitzada per presentar a la part esquerra, el dia i el mes i a la dreta, el titular.

Per poder fer-ho així s'ha hagut de crear un arxiu *xml*, que anomenem *fila_data*, amb l'estructura de la fila. Cada fila conté un *LinearLayout* horitzontal i dins d'aquest, un altre *LinearLayout* vertical i un *TextView* corresponent al titular. Dins del *LinearLayout* vertical hi haurà dos *TextView* corresponents al dia i mes (Figura 55).

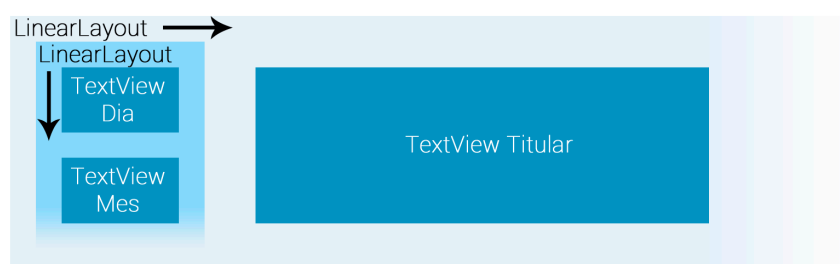


Figura 55. Estructura XML de la fila personalitzada per a la llista de Agenda i Notícies.

Creació de la llista personalitzada.

Per poder mostrar una llista com la que es vol s'ha creat l'estructura de cada fila i l'estructura global, com s'ha vist en l'apartat d' "Estructura". Ara falta emplenar cada element. Es crea la classe *Publicacio* que cada objecte guardarà els elements que es necessiten amb els seus mètodes per obtenir i canviar aquestes variables. Cada objecte *Publicacio* que sigui capturat del document *rssEps.xml*, que és el nom amb que es guardarà a memòria interna el contingut del RSS, formarà una línia de la llista.

```
public class Publicacio {  
    private String mes;           //<summary>  
    private String dia;          //<summary>  
    private String descripcio;   //<title>  
    private String link;         //<link>  
    private String tipus;        //<category> (notícies o agenda)  
    ...  
}
```

Per poder assignar cada variable de l'objecte *Publicacio* a un element de l'estructura de la línia s'ha d'implementar un adaptador. La classe s'ha anomenat *PublicacioAdapter* que hereta *ArrayAdapter<...>*, ja que es crea una llista de publicacions (anomenada *publicacions*) per injectar-la al *ListView*. Solament es necessita sobre-escriure els mètodes heretats constructor, *PublicacioAdapter(...)* que servirà per poder crear un element *PublicacioAdapter*, i *getView()* per injectar a cada fila de la *ListView* els valors d'un element *Publicacio* de la llista de publicacions [2]. Per executar l'adaptador es fa amb el següent codi al fil principal de la interfície:

```
PublicacioAdapter adapter = new PublicacioAdapter(this,  
                                                    R.layout.fila_data, publicacions);  
ListView list = (ListView) findViewById(R.id.llistaAgenda);  
list.setAdapter(adapter);
```

Filtrat del document *rssEps.xml*.

Com ja hem dit abans, es realitzarà la descàrrega del contingut de la web RSS²⁰ i es guardarà en memòria interna del dispositiu, cosa que s'explica després en l'apartat "Emmagatzematge i lectura en memòria del document *rssEps.xml*". Però quan ja es té el document guardat s'ha d'interpretar per extreure les dades que interessin i crear la llista de publicacions. Per fer-ho es crea la classe *BlogEpsXmlParser*. El document RSS té la següent estructura:

```
<?xml version="1.0" encoding="UTF-8"?><feed
xmlns="http://www.w3.org/2005/Atom"
...>
<title type="text">...</title>
<subtitle type="text">...</subtitle>
<updated>2014-07-04T10:29:19Z</updated>
<generator ...>...</generator>
<link ... />
<id>...</id>
<link.../>
<entry>
  <author>
    <name>...</name>
  </author>
  <title type="html"><![CDATA[Lectures Treball Final de
                                Grau/Màster]]></title>
  <link rel="alternate" type="text/html"
        href="http://eps.blogs.udl.cat/2014/07/04/
        lectures-treball-final-de-graumaster-5/" />
  <id>...</id>
  <updated>...</updated>
  <published>...</published>
  <category scheme="http://eps.blogs.udl.cat" term="Agenda"/>
  <category scheme="http://eps.blogs.udl.cat"
            term="Treball/Projecte Final Titulació" />
  <summary type="html"><![CDATA[7 Juliol]]></summary>
  <link rel="replies" .../>
  <link rel="replies" .../>
  <thr:total>0</thr:total>
</entry>
<entry>
  <author>
    <name>...</name>
  </author>
  <title type="html"><![CDATA[Acte de lliurament de les
                                Orles Acadèmiques de l'Escola
                                Politècnica Superior]]></title>
  <link rel="alternate" type="text/html"
        href="http://eps.blogs.udl.cat/2014/07/04/acte-
        de-lliurament-de-les-orles-acadèmiques-de-leps-2/"
  />
```

²⁰ <http://eps.blogs.udl.cat/feed/atom>

```

<id>...</id>
<updated>...</updated>
<published>...</published>
<category scheme="http://eps.blogs.udl.cat" term="Notícies. EPS" />
<summary type="html"><![CDATA[4 Juliol]]></summary>
<link rel="replies" .../>
<link rel="replies" .../>
<thr:total>0</thr:total>
</entry>
...
</feed>

```

En el text del RSS s'ha marcat les etiquetes en diferents colors que indiquen la lectura d'aquestes. L'etiqueta groga són les entrades que en la funció `readChannel()` s'invocarà a la funció `readEntry()`. Les verdes són les que busca la funció `readChannel()` i quan trobarà una d'elles s'executarà la funció corresponent per recuperar el contingut. L'etiqueta marcada en blau s'utilitza en la funció `readUpdate()` per comprovar la versió del document RSS. Totes les etiquetes vermelles són les que no ens interessin i el parser en farà la lectura i invocarà la funció `skip()` per passar a la següent. La funció que s'executa per iniciar l'operació és `.parse()`. Únicament crea un element de la classe `XmlPullParser`, que és com un cursor per llegir el document i es passarà per paràmetre a les següents funcions que s'invocaran per tal de seguir tot el document sense deixar cap element sense comprovar. S'executa el mètode `readChannel(...)`, que realitzarà la lectura de tot el document, i es tanca el `InputStream`.

```

public ArrayList<Publicacio> parse(InputStream in, String tipus)
throws XmlPullParserException, IOException {
    try {
        XmlPullParser parser = Xml.newPullParser();
        parser.setFeature(XmlPullParser.
            FEATURE_PROCESS_NAMESPACES, false);
        parser.setInput(in, null);
        parser.nextTag();
        return readChannel(parser, tipus);
    } finally {
        in.close();
    }
}

```

La funció *nextTag()* fa córrer el punt de lectura fins a la següent etiqueta. Solament s'atura en les etiquetes d'obertura (<...>) i tancament (</...>) de forma que es pot anar avançant en el text fins a les etiquetes desitjades.

La funció *readChannel()* busca l'etiqueta <entry> que és el que marca l'inici d'una publicació. Per fer-ho es crea un bucle que s'atura en les etiquetes d'obertura . Si aquesta etiqueta és <entry> executa *readEntry()* sinó executa *skip()*, el qual busca l'etiqueta de tancament d'aquest element ignorant totes les etiquetes dins aquesta.

```
private ArrayList<Publicacio> readChannel(XmlPullParser parser,
String tipus) throws XmlPullParserException, IOException {
    ...
    while (parser.next() != XmlPullParser.END_TAG) {
        if (parser.getEventType() != XmlPullParser.START_TAG) {
            continue;
        }
        String name = parser.getName();
        if (name.equals("entry")) {
            Publicacio publicacio = readEntry(parser);
            if (publicacio.getTipus().equals(tipus)) {
                publicacions.add(publicacio);
            }
        } else {
            skip(parser);
        }
    }
    return publicacions;
}
```

En aquest punt es selecciona el tipus de publicació desitjada. Tant les publicacions de Notícies com d'Agenda estan incloses en un mateix document. Es diferencien per l'etiqueta <category...> i l'atribut *term="..."*. Com que aquesta classe s'utilitza tant en Agenda com en Notícies, per poder fer la separació s'ha de passar la variable *Tipus* en el moment de fer la crida al *parser*. Després d'executar la funció *readEntry()* es comprova que el valor de *Tipus* que s'ha passat coincideixi amb el valor de l'atribut *term*. En cas que coincideixi s'afegeix la publicació a la llista de publicacions que retornarà *readChannel()*, sinó es continua amb una altra volta al bucle.

```
if (publicacio.getTipus().equals(tipus)) {  
    publicacions.add(publicacio);    //afegim publicació a la llista  
}
```

La funció *readEntry()* retorna un element *Publicacio* per tal que *readChannel()* pugui crear la llista de publicacions, que aquesta envia a *parse()*. Per fer-ho s'executa un bucle que finalitza en el moment que detecta una etiqueta de tancament, que serà sempre *</entry>*. A cada volta comprova el nom de l'etiqueta i si és una de les que es busca s'executa la funció corresponent que separarà la informació correcta.

```
private Publicacio readEntry(XmlPullParser parser) throws  
XmlPullParserException, IOException {  
    Publicacio publicacio = new Publicacio();  
    while (parser.next() != XmlPullParser.END_TAG) {  
        if (parser.getEventType() != XmlPullParser.START_TAG) {  
            continue;  
        }  
        String name = parser.getName();  
        if (name.equals("summary")) {  
            readData(parser);  
            publicacio.setDia(dia);    //dia variable global  
            publicacio.setMes(mes);    //mes variable global  
        } else if (name.equals("title")) {  
            descripcio = readDescription(parser);  
            publicacio.setDescripcio(descripcio);  
        } else if (name.equals("link")) {  
            String text = readLink(parser);  
            if (text != "-1") {  
                publicacio.setLink(text);  
            }  
        } else if (name.equals("category")) {  
            String tipus = readCategory(parser);  
            if (tipus != "-1"){  
                publicacio.setTipus(tipus);  
            }  
        } else { skip(parser); }  
    }  
    return publicacio;  
}
```

Cada variable que es vol guardar de l'objecte *Publicacio* està emmagatzemada de diferent manera entre les etiquetes del document *xml*, per tant, es necessita una funció per a cadascuna. Si es comprova l'estructura del document *RSS* que s'ha presentat abans es veu que, per exemple, no és el mateix recuperar el títol de la publicació:

```
<title type="html"><![CDATA[Acte de lliurament de les Orles
Acadèmiques de l'Escola Politècnica
Superior]]></title>
```

Que el vincle de la notícia:

```
<link rel="alternate" type="text/html"
href="http://eps.blogs.udl.cat/2014/07/04/acte-de-lliurament-de-les-
orles-acadèmiques-de-leps-2/" />
```

Per obtenir el dia i el mes s'ha d'extreure de l'etiqueta `<summary...>` en format `CDATA` i després s'haurà de separar ja que en la cadena hi ha el dia i el mes separats per un espai.

```
<summary type="html"><![CDATA[4 Juliol]]></summary>
```

Com que la data ens ve donada dins de la cadena `<![CDATA[...]]>` s'ha de executar la funció `nextToken()` en lloc de `nextTag()`. Totes dues funcions són molt similars però `nextToken` permet discriminar els caràcters especials que envolten la data i mitjançant la variable `CDSECT`, en realitzar un `getText()`, ens retorna el text de l'interior.

Quan ja s'ha extret la data es requereix la utilització de la classe `StringTokenizer` que permet dividir un text en funció de l'element separador que es vulgui. Per defecte, si no se li indica cap paràmetre com a separador utilitza l'espai i ja va bé perquè sol tenim el dia i el mes separat per un espai.

```
private void readData (XmlPullParser parser) throws IOException,
    XmlPullParserException {
    parser.require(XmlPullParser.START_TAG, ns, "summary");
    String result = "";
    StringTokenizer st;
    if (parser.nextToken() == XmlPullParser.CDSECT) {
        result = parser.getText();
        st = new StringTokenizer(result);
        dia = st.nextToken();           //dia variable global
        mes = st.nextToken();          //mes variable global
    };
    parser.nextTag(); }
```

Per obtenir la descripció s'ha de fer un procés similar al de la data ja que també ve donat amb format `<![CDATA[...]]>`.

```
<title type="html"><![CDATA[Acte de lliurament de les Orles  
Acadèmiques de l'Escola Politècnica  
Superior]]></title>
```

El codi de la funció és el mateix que `readData()` sense la separació de les paraules.

```
private String readDescription(XmlPullParser parser) throws  
IOException, XmlPullParserException {  
    parser.require(XmlPullParser.START_TAG, ns, "title");  
    String result = "";  
    if (parser.nextToken() == XmlPullParser.CDSECT) {  
        result = parser.getText();  
    };  
    parser.nextTag();  
    return transformaText(result);  
}
```

Un punt que s'ha de tenir en compte és que el text ve en format HTML i els caràcters especials en el moment de mostrar-los en pantalla no els transforma. Per canviar el format de text s'ha utilitzat la següent funció que retorna el text en format que sí accepta Android²¹.

```
private String transformaText(String result) {  
    CharSequence styledText = Html.fromHtml(result);  
    return styledText.toString();  
}
```

Per la obtenció del vincle de la Publicació el procés és diferent.

```
<link rel="alternate" type="text/html"  
href="http://eps.blogs.udl.cat/2014/07/04/acte-de-lliurament-de-les-  
orles-acadèmiques-de-leps-2/" />  
<link rel="replies".../>  
<link rel="replies".../>
```

²¹ <http://developer.android.com/guide/topics/resources/string-resource.html>

El valor que s'ha de recuperar és l'atribut *href* i s'ha de comprovar que és el vincle correcte ja que hi ha d'altres etiquetes `<link.../>` però amb l'atribut *rel="replies"*. Per fer-ho es comprova el valor de *rel* i si és *"alternate"* es procedeix a recuperar el valor de *href* i es retorna a *readEntry* perquè ho inclogui en la *Publicacio*. En cas que no ho sigui es retorna *"-1"* per evitar que en *readEntry* sobre-escrigui el valor de la variable *link* de la *Publicacio*.

```
private String readLink(XmlPullParser parser) throws IOException,
XmlPullParserException {
    String link = "link en blanc";
    String relType = parser.getAttributeValue(null, "rel");
    if (relType.equals("alternate")) {
        link = parser.getAttributeValue(null, "href");
        parser.nextTag();
        return link;
    }
    parser.next();
    return "-1";
}
```

Per finalitzar, l'obtenció del tipus de publicació. S'ha de tenir en compte que no solament hi ha una etiqueta `<category.../>` perquè a part de si és *"Agenda"* o *"Notícies. EPS"* hi ha altres classificacions com es veu en el següent codi.

```
<category scheme="http://eps.blogs.udl.cat" term="Agenda"/>
<category scheme="http://eps.blogs.udl.cat" term="Treball/Projecte
Final Titulació" />
```

Sol interessa si és *"Agenda"* o *"Notícies. EPS"* per poder incloure-ho a la llista de la secció d'Agenda o a la de Notícies. Per tant, es soluciona realitzant una comprovació de l'atribut *term* en cada lectura de l'etiqueta `<category.../>`. Si és un d'aquests dos retornem el tipus sinó retornem *"-1"*.

```
private String readCategory(XmlPullParser parser) throws
IOException, XmlPullParserException {
    parser.require(XmlPullParser.START_TAG, ns, "category");
    String termType = parser.getAttributeValue(null, "term");
    if (termType.equals("Notícies. EPS") ||
        termType.equals("Agenda")) {
        parser.nextTag();
        return termType;
    }
}
```

```
    } else {  
        parser.nextTag();  
        return "-1";  
    }  
}
```

En l'estructura del document passada anteriorment (p. 83) s'ha marcat en blau l'etiqueta `<updated>`. Aquesta etiqueta marca la data de publicació d'aquest document i serà necessari més endavant realitzar una lectura del *xml* únicament per buscar aquest valor. Per fer-ho s'ha implementat una funció amb nom *updatedDate()* que retornarà la data en text. És una adaptació de la funció *parse()*.

```
public String updatedDate (InputStream in) throws  
XmlPullParserException, IOException {  
    boolean actualitzat = false;  
    XmlPullParser parser = Xml.newPullParser();  
    parser.setFeature(XmlPullParser.FEATURE_PROCESS_NAMESPACES,  
                                                              false);  
  
    parser.setInput(in, null);  
    parser.nextTag();  
    String dataDocument = readUpdate(parser);  
    return dataDocument;  
}
```

La funció *readUpdate(...)* és molt similar a *readChannel(...)* adaptada per a què busqui sol la primera etiqueta `<updated>`. Amb un booleà (*pendentUpdate*) marquem la sortida del bucle perquè si s'analitzés tot el document es sobre-escriuria la data amb l'etiqueta `<updated>` de les publicacions. D'aquesta forma també s'escurça la lectura solament fins al punt on ens interessa estalviant traspàs de dades i temps.

```
private String readUpdate (XmlPullParser parser) throws IOException,  
XmlPullParserException {  
    String data = "";  
    boolean pendentUpdated = true;  
    parser.require(XmlPullParser.START_TAG, ns, "feed");  
    while (pendentUpdated) {  
        parser.next();  
        if (parser.getEventType() != XmlPullParser.START_TAG) {  
            continue;  
        }  
        String name = parser.getName();  
        if (name.equals("updated")) {  
            if (parser.next() == XmlPullParser.TEXT) {
```

```

        data = parser.getText();
        parser.nextTag();
        pendentUpdated = false;
    }
    } else {
        skip(parser);
    }
}
return data;
}

```

La realització de la classe *BlogEpsXmlParser* s'ha basat en la informació donada al respecte de Parsers XML en la web de *Android Developers*²² transformant el codi per tal que ens sigui útil.

Descarrega del document *xml* i mostrar la interfície.

En aquest cas també ens hem guiat per l'aplicació que *Android Developers* ens posa com a exemple, *NetworkActivity*. En la primera de les seccions descriu la realització d'una connexió a un RSS per mostrar totes les entrades. No és exactament el mateix ja que en aquest cas es vol realitzar la descàrrega del text.

En el fil principal de Agenda o Notícies es realitza el següent procediment:

```

public void onCreate(Bundle savedInstanceState){
    super.onCreate(savedInstanceState);
    setContentView(R.layout.agenda);           //agenda o notícies
    ...
    estatActualitzacio =
        getResources().getString(R.string.pendentUpdate);
    modificarDataActualitzacio(estatActualitzacio); //1
    inicialitzarLlista();                          //2
    actualitzarLlista ();                          //3
    list.setOnItemClickListener(new OnItemClickListener() {
        ...
    });
}

```

Els tres passos són (1) modificar el *TextView* superior per tal d'indicar l'estat en que es troba la llista, (2) inicialitzar la llista des de memòria i, per últim, comprovar si el document XML de la memòria està actualitzat. S'inicialitza la

²² <https://developer.android.com/training/basics/network-ops/xml.html>

l·lista abans de realitzar l'actualització perquè l'actualització requereix crear una tasca asíncrona per connectar a la xarxa i, si la versió és més recent, descarregar el document. Això comporta un temps d'espera que, en funció de la connexió, pot ser de 2 segons a un temps indeterminat. També es té en compte que l'actualització d'aquest document no és diari. Durant el temps que s'ha estat realitzant aquest projecte les actualitzacions s'han produït cada dos o tres dies.

Donat que el *TextView* superior es modifica varies vegades, s'ha implementat el codi en una petita funció *modificarDataActualitzacio(String)* i el mateix passa amb les tres línies que serveixen per inflar la l·lista en pantalla, que s'ha agrupat en *mostrarLlista()* i s'utilitzen dins la funció *inicialitzarLlista()*.

```
private void mostrarLlista () {
    PublicacioAdapter adapter = new PublicacioAdapter(this,
                                                    R.layout.fila_data, publicacions);
    list = (ListView) findViewById(R.id.llistaAgenda);
    list.setAdapter(adapter);
}

private void modificarDataActualitzacio (String text) {
    TextView ultimaActualitzacio = (TextView)
    findViewById(R.id.ultimaActA);
    ultimaActualitzacio.setText(text);
}
```

Com a primer punt, amb la funció *modificarDataActualització()*, es crea el *TextView*, s'inicialitza i s'associa amb l'element de la interfície que correspon mitjançant *findViewById(id)*. *ultimaActA* és el nom clau que se li ha donat al *TextView* de la secció d'agenda. Després mitjançant *.setText(...)* se li assigna el nou valor.

En el segon punt, inicialitzar la l·lista, es crea l'element de *BlogEpsXmlParser* i el canal de transmissió de dades *InputStream*. L'execució s'ha d'incloure dins un *try{...}* perquè totes dues execucions poden retornar excepcions i s'han de capturar en cas que passi. Si fos així es modificaria el text de sobre de la l·lista indicant l'error.

```

private void inicialitzarLlista () {
    BlogEpsXmlParser blogEpsXmlParser = new BlogEpsXmlParser();
    InputStream is = null;
    try {
        is = openFileInput(FILENAME);
        publicacions = blogEpsXmlParser.parse(is, tipus);
        is.close();
    } catch (IOException e) {          //Recull també FileNotFoundException
        estatActualitzacio =
            getResources().getString(R.string.IOException);
    } catch (XmlPullParserException e) {
        estatActualitzacio = getResources()
            .getString(R.string.XmlPullParserException);
    }
    mostrarLlista();
    modificarDataActualitzacio(estatActualitzacio);
}

```

El tercer punt és actualitzar el document guardat en memòria. Per fer-ho es comprova la connexió que es disposa i, si es pot realitzar la connexió, s'executa l' *AsyncTask* amb la comanda *new ActualitzaXML().execute(URL)*. Després de llençar l'actualització s'indica la data en que s'ha realitzat aquesta connexió en el text informatiu de sobre de la llista o, en cas que no hi hagi connexió, així s'indicarà també.

```

private void actualitzarLlista () {
    if (connected()) {
        new ActualitzaXML().execute(URL);
        Calendar avui = Calendar.getInstance();
        DateFormat formatter = new SimpleDateFormat("dd-MM-yyyy");
        modificarDataActualitzacio(getResources().
            getString(R.string.ultimaActualitzacio)+
            formatter.format(avui.getTime()));
    } else {
        modificarDataActualitzacio(getResources()
            .getString(R.string.noConnexioServer));
    }
}

```

La funció *connected()* retorna *true* si es pot realitzar la connexió²³. Primer comprova que el dispositiu tingui connexió i si és així comprova la configuració de les preferències. Si tot és correcte retorna *true*.

²³ <http://developer.android.com/training/basics/network-ops/connecting.html>

```
private boolean connected() {
    ConnectivityManager connMgr = (ConnectivityManager)
        getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo activeInfo = connMgr.getActiveNetworkInfo();
    if (activeInfo != null && activeInfo.isConnected()) {
        if ((sPref.equals(ANY)) || (sPref.equals(WIFI) &&
            (activeInfo.getType() == ConnectivityManager.TYPE_WIFI)))
        {
            return true;
        }
    }
    return false;
}
```

La descàrrega de dades es realitza amb un fil paral·lel en *background* per dos motius. El primer és no retardar l'aplicació ja que si es fes la descàrrega en el fil principal podria tardar un temps indeterminat en realitzar la càrrega de la interfície, cosa que donaria la sensació de lentitud. I el segon és que si la connexió fos prou lenta i es tardes més de cinc segons Android donaria un error indicant que l'aplicació no respon²⁴.

Una *AsynkTask* és una execució en un fil secundari d'una acció transparent a l'usuari. Tal com s'indica a la web d' *Android Developers*²⁵ està pensat per tasques que durin poca estona ja que per realitzar tasques llargues hi ha altres solucions com són *Executor*, *ThreadPoolExecutor* i *FutureTask*. Permet la implementació de fins a quatre mètodes:

- *onPreExecute()*. Accions a realitzar abans de la funció en segon pla com pot ser la creació d'una barra de progrés.
- *doInBackground()*. És on es desenvolupa tota la funció en segon pla.
- *onProgressUpdate()*. Permet incloure accions a realitzar mentre s'està duent a terme *doInBackground* i serviria, per exemple, per incloure una animació que indica que s'està realitzant una descàrrega.
- *onPostExecute()*. Accions a realitzar un cop finalitzada la tasca en segon pla. En cas de l'aplicació de l'EPS s'utilitza per tornar a inicialitzar la llista amb les noves publicacions.

²⁴ <http://www.sgoliver.net/blog/?p=3099>

²⁵ <http://developer.android.com/reference/android/os/AsyncTask.html>

La creació d'una classe com a *AsyncTask* implica sobre-escriure almenys el mètode *doInBackground(...)* i en aquest cas, es necessitarà també *onPostExecute(...)* per inflar la llista de nou si s'ha realitzat alguna actualització.

```
private class ActualitzaXML extends AsyncTask<String, Void, String>{

    @Override
    protected String doInBackground(String... urls) {
        try {
            if (updated(urls[0])){
                return "updated";
            } else {
                return loadXmlFromNetwork(urls[0]);
            }
        } catch (IOException e) {
            return getResources().getString(R.string.
                                           connection_error);
        } catch (XmlPullParserException e) {
            return getResources().getString(R.string.xml_error);
        }
    }

    @Override
    protected void onPostExecute(String result) {
        if (result.equals("updated")==false) {
            inicialitzarLlista();
        }
    }
}
```

En la funció *doInBackground(...)* es comprova amb *updated(...)* si el document està actualitzat. Si ho està retorna "true" i en *onPostExecute(...)* no realitza cap acció. En cas que no estigui actualitzat crida la funció *loadXmlFromNetwork(...)* que realitzarà la connexió i la descàrrega.

```
private boolean updated (String urlString) {
    String dataAvui = "";
    String dataDocGuardat = "";
    String dataDocRSS = "";
    BlogEpsXmlParser blogEpsXmlParser = new BlogEpsXmlParser();
    InputStream is = null;
    Calendar avui = Calendar.getInstance();
    SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");
    dataAvui = df.format(avui.getTime());
    try {
        is = openFileInput(FILENAME);          //connexió al arxiu intern
        dataDocGuardat = blogEpsXmlParser.updatedDate(is);
    } catch (IOException e) {
        estatActualitzacio =
    }
```

```
        getResources().getString(R.string.IOException);
    } catch (XmlPullParserException e) {
        estatActualitzacio = getResources()
            .getString(R.string.XmlPullParserException);
    }
    if (dataAvui.contains(dataDocGuardat)) {
        return true;
    } else {
        try{
            is = downloadUrl(urlString);        //connexió a la web RSS
            dataDocRSS = blogEpsXmlParser.updatedDate(is);
        } catch (IOException e) {
            estatActualitzacio = getResources()
                .getString(R.string.IOException);
        } catch (XmlPullParserException e) {
            estatActualitzacio = getResources()
                .getString(R.string.XmlPullParserException);
        }
        return dataDocGuardat.equals(dataDocRSS);
    }
}
```

En el primer *try {...}* s'obre la connexió amb el document guardat en memòria del dispositiu i compara la data del document amb la data actual. Es fa la comparació amb el mètode *.contains(...)* perquè no interessa l'hora ja que no s'actualitza dues vegades en el mateix dia i es vol evitar realitzar una connexió cada cop que s'entra en la secció. El format de la data actual que s'ha obtingut és "2014-07-04" i el que hi ha al document *xml* és "2014-07-04T10:29:19Z". En cas que la comparació coincideixi, dins un segon *try {...}* s'obre el canal amb la web RSS i s'obté la data d'actualització. En aquest cas, si que es realitza la comparació directa mitjançant *.equals(...)* perquè tots dos documents tenen el mateix format de data i han de ser iguals.

La funció *loadXmlFromNetwork(...)* realitza una lectura línia a línia del RSS a un *String* que després es guarda en memòria interna. Sempre es guarda amb el mateix nom, per tant, sempre s'eliminarà el document antic per evitar acumulació de arxius *xml* en memòria.

```
private String loadXmlFromNetwork(String urlString) throws
XmlPullParserException, IOException {
    InputStream stream = null;
    String text = "";
    BufferedReader bufferReader = null;
    try {
        stream = downloadUrl(urlString);
```



```

        bufferReader = new BufferedReader(new
                                   InputStreamReader(stream));
        String StringBuffer = bufferReader.readLine();
        while (StringBuffer != null) {
            text += StringBuffer+"\n";
            StringBuffer = bufferReader.readLine();
        }
    } finally {
        if (stream != null) {
            stream.close();
        }
    }
    bufferReader.close();
    try {
        FileOutputStream fos = openFileOutput("rssEps.xml",
        Context.MODE_PRIVATE);
        fos.write(text.getBytes());
        fos.close();
    } catch (IOException e){
        estatActualitzacio =
            getResources().getString(R.string.IOException);
    }
    return text;
}

```

Emmagatzematge i lectura en memòria del document *rssEps.xml*.

Donat que el document *xml* que s'obté ocupa molt poc espai es guarda directament en memòria interna²⁶. Les comandes són senzilles i s'utilitzen en un parell de llocs.

La lectura es realitza en la funció *inicialitzarLlista(...)* i consta de la creació d'un element de la classe *InputStream* que serveix de canal per recuperar les dades, l'obertura de l'arxiu mitjançant *openFileInput(...)* i tancant el canal amb *.close()*. En la web consultada s'indica que la lectura es crea amb un element *FileInputStream* però aquesta és una sub-classe de *InputStream* i per realitzar l'anàlisi del document amb el *parser* es necessita un objecte tipus *InputStream*.

```

...
InputStream is = null;
...
is = openFileInput(FILENAME);
...
is.close();

```

²⁶ <http://developer.android.com/guide/topics/data/data-storage.html#filesInternal>

La gravació en memòria s'utilitza en la funció *loadXmlFromNetwork(...)* i consta de la creació d'un element de la classe *FileOutputStream*, guardant el document amb *.write(...)* i tancant el canal amb *.close()*.

```
...
FileOutputStream fos = openFileOutput("rssEps.xml",
                                   Context.MODE_PRIVATE);
fos.write(text.getBytes());
fos.close();
...
```

La funció *downloadUrl(...)* és la mateixa que apareix en la aplicació exemple que proporciona Android com a tutorial *NetworkActivity*²⁷.

8.2.4. Preferències.

L'apartat de preferències s'ha realitzat mitjançant la classe *PreferenceActivity*²⁸. La utilització d'aquesta classe és molt senzilla, ja que ella mateixa s'encarrega de gestionar les preferències en un arxiu XML amb la relació clau - valor i guardant les modificacions. Per implementar una interfície de preferències amb aquesta classe és necessari crear primer una estructura en un arxiu XML en la carpeta *res/xml/preferences*. L'arxiu XML conté una etiqueta base *<PreferenceScreen ...>* on s'inclouran els diferents elements corresponents a cada variable.

Existeixen diferents tipus d'elements. Cada element pot tenir diferents atributs depenent de la informació que necessiten i opcions que ofereixen.

- *PreferenceCategory*. Separador que serveix per agrupar els elements del mateix tipus o utilitat.
- *CheckBoxPreference*. Permet marcar o desmarcar l'element corresponent a un booleà amb forma de casella.
- *SwitchPreference*. S'utilitza igual que el *checkBox* però la visualització és en forma d'interruptor.

²⁷ <http://developer.android.com/training/basics/network-ops/connecting.html>

²⁸ <http://www.sgoliver.net/blog/?p=2149>

- *EditTextPreference*. Permet guardar una variable *String*. En el moment que es selecciona aquest element, s'obre una finestra de diàleg amb un camp de text on es pot escriure el que es vulgui.
- *ListPreference*. Permet seleccionar d'una llista un únic element mitjançant una finestra on surten tots els elements.
- *MultiSelectListPreference*. Igual que l'element anterior, obre un diàleg on surt un conjunt d'opcions però en aquest cas es pot marcar més d'una opció o inclús cap.
- *PreferenceScreen*. És l'element base del document de preferències però si se n'inclou una altra entrada ens permet obrir en una altra finestra tots els elements que hi hagi dins l'etiqueta.

Estructura.

Els punts, previstos en els prototips, que interessa guardar per part dels usuaris és la selecció d'assignatures que es mostraran al calendari, idioma, mida de les fonts, selecció de la pantalla d'inici, configuració de les notificacions del calendari i configuració de la connexió per descarrega.

La selecció d'assignatures s'ha configurat de forma que s'obre una nova interfície per seleccionar la titulació i després una altra amb les assignatures per marcar-les. Si s'hagués fet tot en una mateixa interfície ens donaria com a resultat una llista llarguíssima d'assignatures que seria difícil de gestionar per part de l'usuari. Per crear una nova interfície s'utilitza l'etiqueta `<PreferenceScreen...>` de nou. Per seleccionar les assignatures s'ha utilitzat `<CheckBoxPreference...>` on es configura *android:key*, que és l'identificador per poder recuperar la configuració en la part de Java, i *android:title*, que és el text de la casella on s'hi veurà el nom de l'assignatura. En les que el nom era massa llarg perquè fos completament visible, s'ha abreviat i s'ha utilitzat l'atribut *android:summary*.

Per separar els cursos s'ha marcat amb l'etiqueta `<PreferenceCategory...>` que únicament requereix introduir el nom.

```
<PreferenceScreen
    android:title="Assignatures"
    android:summary="Selecciona les assignatures que vols que
es sincronitzin al teu calendari">
    <PreferenceScreen
        android:key="screen_arquitectura"
        android:title="Grau en Arquitectura Tècnica">
        <PreferenceCategory
            android:title="PRIMER CURS">
            <CheckBoxPreference
                android:key="101402"
                android:title="Física"/>
            ...
            <CheckBoxPreference
                android:key="101414"
                android:title="Història de la Construcció"/>
        </PreferenceCategory>
        <PreferenceCategory
            android:title="SEGON CURS">
            <CheckBoxPreference
                android:key="101412"
                android:title="Materials 2"/>
            ...
            <CheckBoxPreference
                android:key="101422"
                android:title="E. obra, Inst. i Mitjans Aux."
                android:summary="Equips d'obra, Instal·lacions
                                i Mitjans Auxiliars"/>
            ...
        </PreferenceCategory>
        ...
    </PreferenceScreen>
    ...
</PreferenceScreen>
```

La part de la selecció de pantalla inicial, l'idioma i de la mida de lletra s'ha realitzat amb una `<ListPreference...>`. A la selecció com a pantalla inicial hem ficat totes les seccions disponibles, és a dir, Agenda, Notícies, Matrícula, Guia Docent i Calendari. Com a idiomes s'ha configurat l'opció de català i una de castellà i en la selecció de mida de lletra s'han configurat tres opcions. En els elements `<ListPreference...>` es configura l'atribut `android:key`, `android:title` i si fa falta una explicació `android:summary` i per determinar els elements de la llista que es mostrarà s'inclou l'atribut `android:entries`, que és la llista dels texts que es veuran per cada element de la llista, i `android:entryValues`, que és una llista paral·lela a la dels texts però amb el codi que s'utilitzarà en la part de programació.

```

<PreferenceCategory
    android:title="@string/category_preferenciesAltres">
    <ListPreference
        android:key="llista_pantalla_inicial"
        android:title="@string/pantalla_inicial"
        android:summary="@string/pantalla_inicial_summary"
        android:entries="@array/PANTALLA_INICIAL"
        android:entryValues="@array/PANTALLA_INICIAL_CODI"/>
    <ListPreference
        android:key="llista_idioma"
        android:title="@string/idioma"
        android:entries="@array/IDIOMA"
        android:entryValues="@array/IDIOMA_CODI"/>
    <ListPreference
        android:key="llista_tamany_lletra"
        android:title="@string/tamany_lletra"
        android:entries="@array/TAMANY"
        android:entryValues="@array/TAMANY_CODI"/>
</PreferenceCategory>

```

La configuració de les notificacions en els prototips de paper es va dissenyar que es podria indicar quin tipus d'avisos rebre. Després d'haver realitzat la implementació de la majoria de seccions s'ha suprimit les notificacions de noves notícies i noves entrades en l'agenda ja que l'actualització de publicacions es realitza en períodes llargs i es comprova cada cop que entres en la secció. S'ha afegit una entrada amb l'opció de marcar l'antelació que es vol per els esdeveniments del calendari.

```

<PreferenceCategory
    android:title="@string/category_notificacions">
    <SwitchPreference
        android:title="@string/notificacions"
    />
    <MultiSelectListPreference
        android:key="llista_tipus_notificacions"
        android:title="@string/tipusNotificacions"
        android:entries="@array/TIPUS_NOTIFICACIONS"
        android:entryValues="@array/TIPUS_NOTIFICACIONS_CODI"
    />
    <ListPreference
        android:key="llista_notificacions"
        android:title="@string/config_notificacions"
        android:entries="@array/TEMPS_NOTIFICACIONS"
        android:entryValues="@array/TEMPS_NOTIFICACIONS_CODI"
    />
</PreferenceCategory>

```

Per últim, la configuració del tipus de xarxa per realitzar les descàrregues es realitza amb un `<ListPreference...>`.

```
<PreferenceCategory
    android:title="@string/category_networks">
    <ListPreference
        android:key="llista_networks"
        android:title="@string/networks"
        android:entries="@array/NETWORKS"
        android:entryValues="@array/NETWORKS_CODI"/>
    </PreferenceCategory>
```

Implementació

La implementació és molt simple. Solament cal crear la classe que hereta *PreferenceActivity* i sobre-escriure el mètode *onCreate()* per poder fer la càrrega del document *xml* i fer les modificacions que corresponguin amb els canvis que es fan.

```
public class SettingsActivity extends PreferenceActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        addPreferencesFromResource(R.xml.preferences);
    }
}
```

Recuperació de les preferències i utilització.

En les seccions d'Agenda i Notícies es necessari consultar les preferències per poder realitzar la descàrrega de les actualitzacions. Es creen variables globals per tal que la utilització en segon pla no doni problemes.

```
...
public static final String WIFI = "Wi-Fi";
public static final String ANY = "Any";
public static String sPref = null;
```

La recuperació de les preferències es realitza mitjançant el mètode *getDefaultSharedPreferences(this)* en un element de la classe *SharedPreferences* i

per recuperar una clau en concret *getString(nomKey, defaultValue)* on es passa per paràmetres el nom clau i, en cas que no existís aquesta clau, el valor per defecte.

```
public void onCreate(Bundle savedInstanceState){
    super.onCreate(savedInstanceState);
    ...
    SharedPreferences sharedPrefs =
        PreferenceManager.getDefaultSharedPreferences(this);
    sPref = sharedPrefs.getString("llista_networks", "Wi-Fi");
}
```

En el moment que s'ha de fer una connexió es realitza una crida a la funció *connected()* per tal que es comprovi l'estat de comunicacions del dispositiu i es compara amb les preferències del usuari.

```
private boolean connected() {
    ConnectivityManager connMgr = (ConnectivityManager)
        getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo activeInfo = connMgr.getActiveNetworkInfo();
    if (activeInfo != null && activeInfo.isConnected()) {
        if ((sPref.equals(ANY)) || (sPref.equals(WIFI) &&
            (activeInfo.getType() == ConnectivityManager.TYPE_WIFI))){
            return true;
        }
    }
    return false;
}
```

Per poder implementar l'accés directe en el moment d'obrir l'aplicació a la secció desitjada per l'usuari, es decideix introduir a l'inici el que s'anomena *Splash*. És una pantalla de presentació que únicament conté una imatge que dura uns pocs segons i on es pot incloure la informació referent a versions, autors, etc. S'inclou un *Splash* per poder afegir el codi necessari que ens recuperi la selecció de l'usuari en preferències. Com en Agenda i Notícies, es realitza la càrrega de la preferència de l'usuari en la variable *sPref* i després, en funció del valor, es crea l'Intent corresponent a la classe indicada que ens obrirà la secció desitjada.

```
SharedPreferences sharedPrefs =
    PreferenceManager.getDefaultSharedPreferences(this);
```

```
sPref = sharedPreferences.getString("llista_pantalla_inicial", "menu");
if (sPref.equals("agenda")) {
    intent = new Intent(this, Agenda.class);
} else if (sPref.equals("noticies")) {
    intent = new Intent(this, Noticias.class);
} else if (sPref.equals("guia")) {
    intent = new Intent(this, Guia.class);
} else if (sPref.equals("matricula")) {
    intent = new Intent(this, Matricula.class);
} else if (sPref.equals("calendari")) {
    intent = new Intent(this, Calendari.class);
} else {
    intent = new Intent(this, MainActivity.class);
}
...
StartScreen.this.startActivity(intent);
```

Per problemes de calendari no s'ha implementat fins al moment el resultat de canviar la mida de fonts o l'idioma i les notificacions i assignatures van lligades a la creació de la secció del calendari.

8.2.5. Pantalla de presentació (*Splash*).

Com ja s'ha comentat, per implementar l'inici de l'aplicació directe a la secció desitjada per l'usuari s'ha creat un *Activity* de presentació. En aquest tipus de pantalles es pot incloure informació relativa a l'aplicació però de moment sol s'ha creat la imatge de fons amb l'emblema de la Universitat i l'Escola Politècnica Superior i un *TextView* indicant que s'està inicialitzant l'aplicació. S'ha triat uns 2 segons de visualització i es passa a la secció.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.splash);
    SharedPreferences sharedPrefs =
    PreferenceManager.getDefaultSharedPreferences(this);
    sPref = sharedPrefs.getString("llista_pantalla_inicial",
                                                                    "menu");

    if (sPref.equals("agenda")) {
        intent = new Intent(this, Agenda.class);
    } else if (sPref.equals("noticies")) {
        intent = new Intent(this, Noticies.class);
    } else if (sPref.equals("guia")) {
        intent = new Intent(this, Guia.class);
    } else if (sPref.equals("matricula")) {
        intent = new Intent(this, Matricula.class);
    } else if (sPref.equals("calendari")) {
        intent = new Intent(this, Calendari.class);
    } else {
        intent = new Intent(this, MainActivity.class);
    }
    new Handler().postDelayed(new Runnable() {
        @Override
        public void run() {
            StartScreen.this.finish();
            StartScreen.this.startActivity(intent);
        }
    }, 2000);
}
```

Com es pot veure la classe és senzilla. L'únic element interessant és el retard que creem amb la funció `Handler().postDelayed(...)`²⁹. Aquesta funció permet incloure un codi (amb la classe *Runnable*) que s'executarà després del temps

²⁹ <http://stackoverflow.com/questions/17961848/android-control-goes-back-to-splash-screen/17961862#17961862>

indicat. Abans d'iniciar la nova *Activity* es finalitza la del *Splash* perquè s'havia observat que cada cop que s'utilitzava el retorn al menú s'accedia al splash i s'havia d'esperar 2 segons per arribar al menú. Si es finalitza amb el mètode *finish()* es soluciona aquest problema.

Per evitar que al prémer el botó de retorn del dispositiu en la pantalla de Menú no obris de nou el Splash es sobre-escriu la funció *onBackPressed()* per tal que finalitzi l'aplicació.

```
@Override
public void onBackPressed() {
    finish();
}
```

8.2.6. Calendari

En tots els prototips el calendari estava previst molt similar al que proporciona Android integrat en els dispositius. Una graella que en funció del tipus de vista que es té seleccionada (Mes, Setmana, Dia i mode Agenda) pren una forma diferent. Després de buscar informació sobre la creació d'una aplicació d'aquest tipus es descarta aquesta idea per dos motius.

Primer que la creació d'aquestes diferents vistes implica la creació d'un volum de codi inassolible en aquest punt i es disposa de poca informació del tractament de calendaris ja que no s'ha trobat en cap dels llibres consultats suficient informació sobre el tema. Sol s'ha trobat informació sobre les classes necessàries en la web de Android en anglès i sense exemples.

I el segon punt va relacionat amb el primer. Per què crear tota aquesta nova aplicació dins de la de l'EPS si ja es disposa de la que ve integrada en el mòbil?

Per tant, s'abandona la idea de crear un calendari amb vista de graella i, després de valorar diferents opcions,



Figura 56. Vista Agenda del calendari dels dispositius Android

es creu que la millor i la que més respectaria la idea inicial seria crear una vista al estil "Agenda" del calendari.

Consisteix en una llista d'esdeveniments agrupats per dies i identificats, segons el tipus de calendari, amb colors. En cada element s'indica el nom de l'esdeveniment, l'hora i el lloc.

Malauradament per motius de calendari no es pot arribar a implementar aquesta opció i es deixarà com a treballs futurs (pàgina 119).

8.2.7. ActionBar

L' *ActionBar* de Android és una barra on es mostra la icona de l'aplicació, el títol i un menú desplegable (menú *Overflow*). La creació de l'*ActionBar* et permet incloure accessos directes i accions en forma d'icona o, en cas que no hi hagi suficient espai o no es vulguin visibles, en el menú desplegable.

Menu.xml

Per a l'aplicació de l'EPS, a banda de la icona i el nom de l'aplicació, s'han considerat necessaris cinc elements.

- Accés a la web. Cada secció té l'opció d'accedir directament a la pàgina web de l'EPS en la secció corresponent.
- Preferències. Accés a la secció de preferències.
- Refrescar. Solament s'utilitza en les seccions de notícies i agenda. Serveix per actualitzar la informació de pantalla cridant a la funció *actualitzarLlista()*.
- Ajuda. Si es selecciona surt una finestra donant indicacions del funcionament de la secció.
- Sobre aquesta aplicació. Igual que la icona d'ajuda, mostra una finestra amb informació sobre l'aplicació.

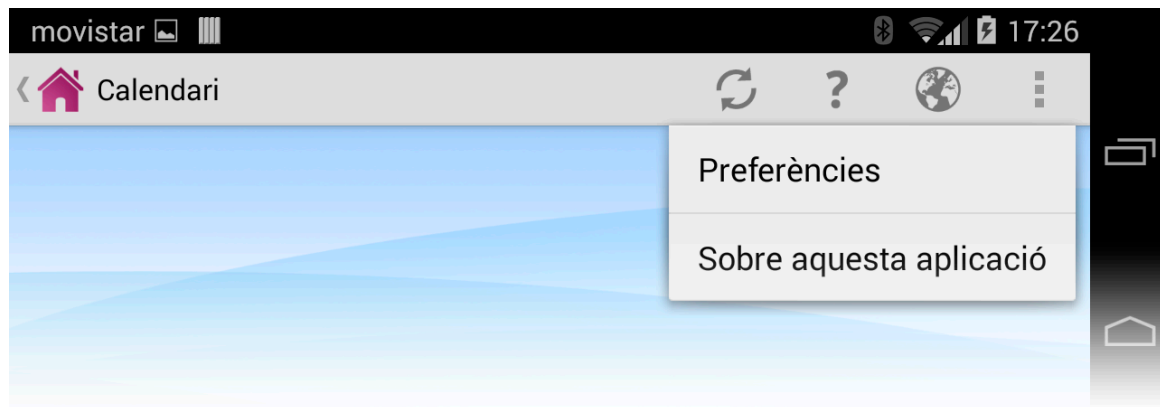


Figura 57. ActionBar en horitzontal on es mostren tots els accessos que creats. Alguns en icona i d'altres en el menú desplegable.

Els elements de la barra es defineix en un arxiu XML a la carpeta `\res\menu\action_bar.xml`. En aquest document es configuren tots els accessos de la barra i s'indica si es vol que siguin visibles com a icona, text, en el menú desplegable o que el mateix sistema s'encarregui de incloure-les o no en funció de l'espai disponible en la barra.

```
<?xml version="1.0" encoding="UTF-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item
    android:id="@+id/refresh"
    android:title="@string/actualitzar"
    android:icon="@drawable/ic_action_refresh"
    android:showAsAction="ifRoom"
  />
  <item
    android:id="@+id/help"
    android:title="@string/help"
    android:icon="@drawable/ic_action_help"
    android:showAsAction="ifRoom"
  />
  <item
    android:id="@+id/web"
    android:title="@string/web"
    android:icon="@drawable/ic_action_web_site"
    android:showAsAction="ifRoom"
  />
  <item
    android:id="@+id/settings"
    android:title="@string/settings"
    android:icon="@drawable/ic_action_settings"
    android:showAsAction="never"
  />
  <item
    android:id="@+id/about"
    android:title="@string/about"
    android:showAsAction="never"
  />
</menu>
```

Com es pot observar en el codi, l'atribut *showAsAction* és "*never*" en preferències i en la informació de l'aplicació, ja que es considera que són accessos menys importants i no és necessari que estiguin com a icona. L'accés a la informació de l'aplicació sol serà accessible des del menú principal i la icona de refrescar sol estarà disponible en les seccions de Agenda i Notícies, ja que són les úniques que hi ha alguna cosa a actualitzar. Com que s'ha creat un únic document de menú es deshabilitaran els que no es desitgen en cada secció. També es pot crear diferents estructures de menú e inflar en cada secció la que li correspon però s'ha decidit realitzar-ho així per simplificar.

Botó retorn / menú i etiqueta de l' Activity.

Com es veu en la Figura 57 la barra consta del nom de la secció, les icones i a l'esquerra del tot hi pot haver la icona de l'aplicació o, com és el cas de l'exemple, el botó de menú habilitat.

El text que es veu en la barra es configura en l'arxiu *AndroidManifest.xml* amb l'atribut *android:label* de cada etiqueta *<activity ...>*.

```
<activity
    android:name="eps.miquel.tfc.Matricula"
    android:label="Matrícula"
    android:parentActivityName="eps.miquel.tfc.MainActivity"
    android:icon="@drawable/ic_action_home" >

</activity>
```

Per activar el botó de retorn al menú consta de dos punts. Primer, com es veu en el codi anterior, es marca amb l'atribut *android:parentActivityName* la classe que es vol activar en el moment que es toca el botó. El segon punt és indicar que l'*ActionBar* té habilitat el botó de retorn. Això es realitza en cada classe que es vol tenir habilitat el botó amb la comanda següent dins de la funció *onCreate()*.

```
getActionBar().setDisplayHomeAsUpEnabled(true);
```

Implementar l' ActionBar en Java.

Per inflar la barra d'acció en l'aplicació s'ha de executar la funció *onCreateOptionsMenu(...)*. Dins aquesta funció s'associa la barra a l'arxiu XML que s'ha creat.

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.action_bar, menu);
    ...
    return true;
}
```

Per deshabilitar les icones que no es vol visibles en cada secció es crea un element *MenuItem*, s'associa per l'*id* a la icona i amb la variable *setVisibility()* i *invalidateOptionsMenu()* s'elimina de la barra.

```
MenuItem item;
item = menu.findItem(R.id.about);
item.setVisible(false);
this.invalidateOptionsMenu();
```

Gestió dels clics en ActionBar.

Per gestionar la selecció dels botons es sobre-escriu el mètode *onOptionsItemSelected(...)*. Amb la comanda *item.getItemId()* es recupera l'id de l'accés seleccionat i s'executa el codi corresponent.

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.settings:
            Intent settings = new Intent(this,
                SettingsActivity.class);
            startActivity(settings);
            return true;
        case R.id.web:
            Intent intent = new Intent(Intent.ACTION_VIEW,
                Uri.parse("http://eps.blogs.udl.cat/"));
            startActivity(intent);
            return true;
        case R.id.refresh:
            Toast.makeText(this, getResources().getString(
                R.string.actualitzant), Toast.LENGTH_SHORT).show();
    }
    return false;
}
```

```
        actualitzarLlista();
        return true;
    case R.id.about:
        aboutDialog();
        return true;
    case R.id.help:
        helpDialog();
        return true;
    default:
        return super.onOptionsItemSelected(item);
    }
}
```

Si es selecciona el boto d'informació sobre l'aplicació o ajuda s'executen dues funcions que mostren un *alertDialog* amb el text corresponent.

```
private void helpDialog() {
    CharSequence text = "text d'ajuda";
    AlertDialog.Builder alert = new AlertDialog.Builder(this);
    alert.setMessage(text);
    alert.setPositiveButton(android.R.string.ok, null);
    alert.show();
}

private void aboutDialog() {
    CharSequence text = "text d'informació";
    AlertDialog.Builder alert = new AlertDialog.Builder(this);
    alert.setMessage(text);
    alert.setPositiveButton(android.R.string.ok, null);
    alert.show();
}
```

L'opció “refrescar” executa la funció *actualitzarLlista()* mostrant un missatge en pantalla i “accés a la web” executa la direcció web indicada segons la secció on s'està en aquell moment i s'obre en el navegador del dispositiu.

8.3. Iconografia

En el disseny d'icones s'ha de tenir en compte les diferents densitats. En la Taula 2 es mostra les mides en *Píxels* per les diferents densitats. En la primera fila es veu la mida en *dp* recomanada per Android per cada tipus d'icona. La primera columna indica la relació d'augment entre les densitats. Com es pot comprovar 1 *dp* equival a 1 *px* en MDPI.

Augments	Nom densitat	DPI	Notificacions (px)		ActionBar (px)		Launcher (px)	Mida mínima visible (px)
			Quadre	Imatge	Quadre	Imatge		
Mida en <i>dp</i> .			24x24	22x22	32x32	24x24	48x48	2
Línia base	MDPI	160	24x24	22x22	32x32	24x24	48x48	2
1,5	HDPI	240	36x36	33x33	48x48	36x36	72x72	3
2	XHDPI	320	48x48	44x44	72x72	48x48	96x96	4
3	XXHDPI	480	72x72	66x66	96x96	72x72	144x144	6
4	XXXHDPI	640	96x96	88x88	128x128	96x96	192x912	8

Taula 3. Mides de les icones en funció de la densitat.

Per poder adaptar l'aplicació a les diferents pantalles es crea una carpeta *\drawable* per a cada densitat creant una versió de cada imatge en cada densitat amb les mides indicades anteriorment.

Les icones et proporcionen una petita informació sobre el contingut per tant, és important dedicar-hi un temps i realitzar varies proves per trobar la definitiva.

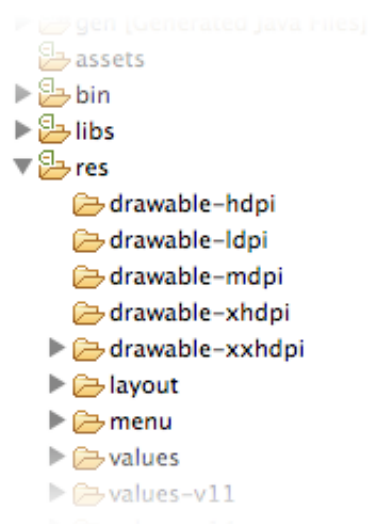


Figura 58. Esquema de les carpetes *res\drawable* en Eclipse.

8.3.1. Icona de l'aplicació (*Launcher*).

La icona de llançament de l'aplicació és una petita representació visual de l'aplicació que es crea. S'ha pensat que en l'escriptori l'usuari tindrà moltes més icones, per tant, s'ha d'intentar realitzar una icona visible en qualsevol fons de pantalla. El disseny d'una silueta diferent amb una mica de profunditat ajuda a visualitzar millor la icona³⁰.

En aquest cas es va buscar algun tipus d'emblema de l'EPS però no hi ha un logotip clar de l'escola. S'ha intentat realitzar una conversió de l'emblema de la UdL amb diferents variacions fins que s'ha triat una icona senzilla que serà visible en diferents densitats sense cap problema i al mateix temps no caldran massa modificacions per realitzar les icones de notifikacions, cosa que reforçarà la identificació d'aquestes.

Ja que s'utilitza part de l'emblema de la Universitat de Lleida no s'entendria que no s'utilitzessin els mateixos colors, per tant s'aprofita el blanc, gris i el lila per donar un color conegut a la icona.



Figura 59. Evolució de la icona de l'aplicació.

³⁰ <http://developer.android.com/design/style/iconography.html>

8.3.2. Icones del menú.

Al dissenyar les icones del menú principal s'ha intentat que hi hagi una relació entre el dibuix i la informació que mostrarà. En el prototip digital s'utilitzen les mateixes icones que proporciona Android per l' *ActionBar* i ja en la versió definitiva s'intenta crear un estil que s'ajusti més a l'esquema de colors i disseny general.



Figura 60. Evolució de les icones del menú.

8.3.3. Icones de la barra d'acció.

Per a l' *ActionBar* només s'ha creat la icona de la casa que significa la tornada a l'inici de l'aplicació (Figura 63). S'ha imitat la icona de la pàgina web i s'ha creat amb les dimensions recomanades per Android. És millor realitzar dibuixos poc carregats amb línies clares. Acabar les vores amb formes molt marcades ja sigui amb punta o suaus i arrodonits. La resta d'icones són les proporcionades per Android i que es poden descarregar gratuïtament (Figura 62).



Figura 61. Captura de la pàgina web de l'EPS de la icona d'inici.

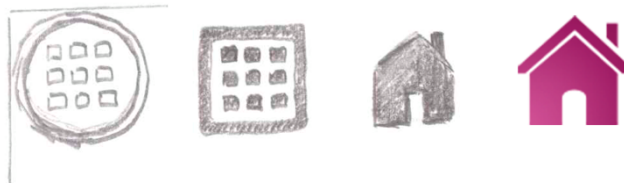


Figura 63. Evolució del disseny de la icona "inici" de la barra d'acció.



Figura 62. Icones que s'han utilitzat en l'aplicació proporcionades per Android.

9. Conclusions

Havent creat l'aplicació de l'Escola Politècnica Superior es compleix l'objectiu principal d'aquest projecte. L'aplicació està operativa totalment en quatre de les cinc seccions que s'havien plantejat en els prototips, assolint també l'objectiu de que fos una aplicació lleugera (amb un baix consum de memòria ja que ocupa solament 2Mb, força inferior a la mitjana de les aplicacions consultades en l'anàlisi de la competència). Per altra banda, per manca de temps, no s'ha assolit el seu llançament o desplegament. Està previst, però, buscar la manera de distribuir-la entre els usuaris posteriorment. De totes formes s'ha inclòs com a treball futur.

Dels objectius funcionals establerts a l'inici del projecte únicament no s'ha completat el canal de comunicació entre usuaris, professors i PAS. Aquest punt s'havia de configurar en la secció de calendari i, com s'ha explicat en l'apartat corresponent, no s'ha pogut realitzar. S'ha detallat tota la informació que es disposa en la part de treballs futurs per tal que es pugui implementar com a futur projecte.

De la resta d'objectius (funcionals o no) fixats en l'anàlisi de requisits, la majoria s'han assolit. Dels funcionals ja s'ha comentat en el paràgraf anterior; respecte els d'usabilitat han estat pràcticament assolits; i, en pel que fa als d'accessibilitat el punt dels colors visualment conflictius també ha quedat correctament solucionat. El tema del reconeixement de veu, el canvi de mida de lletra i el canvi d'idioma són els punts no assolits en aquest projecte però es plantejaran com a treballs futurs per tal que en una hipotètica continuació d'aquest projecte hi fos contemplat.

Tot i això, el desconeixement de la tecnologia Android abans d'iniciar la programació, molt probablement ha induït a buscar uns objectius més assequibles i ha influenciat lleugerament en el procés de disseny de l'aplicació. És possible que si s'hagués fixat uns objectius més ambiciosos podrien no

haver-se assolit, com és el cas de la secció de calendari o el fet de no implementar el desplaçament lateral entre seccions.

Personalment, opino que la realització d'aquest projecte ha sigut per mi una experiència molt enriquidora a nivell d'adquisició de nous coneixements i de pràctica de desenvolupaments actuals.

A part que ha suposat l'entrada a un món totalment nou per mi, com és el disseny i la programació per a dispositius mòbils (cal recordar que quan jo vaig estudiar les matèries, tot aquest món era encara inexistent), l'aplicació pràctica del model de procés MPlu+a m'ha ajudat enormement a entendre la importància de la realització d'un anàlisi de requisits exhaustiu i ben documentat i de la gran font d'informació que suposen els testos amb usuaris. Crec que focalitzar millor les preguntes de l'enquesta o, realitzar abans alguna sessió grupal amb usuaris, hagués ajudat a determinar uns objectius per l'aplicació més concrets i clars. La manera més ràpida i eficaç de conèixer els requisits d'una aplicació es de la mà dels propis usuaris.

10. Treballs futurs

Durant la realització de l'aplicació han quedat idees pendents de realitzar i n'han anat sorgint d'altres que serien bastant interessants. En els punts següents es realitza una descripció de possibles ampliacions o canvis en l'aplicació realitzables en propers treballs.

10.1. Calendari personalitzat.

El punt més atractiu del projecte era la possibilitat de crear aquest canal de comunicació entre alumnat, professorat i PAS a través dels esdeveniments del calendari i notificacions.

Consisteix en sincronitzar els calendaris de les assignatures que es vulgui i el calendari de secretaria on s'hi inclou festius, períodes d'interès, etc... en el dispositiu de l'usuari. Cada tipus d'usuari té una funció diferent:

- Els alumnes solament poden visualitzar els calendaris que ells mateixos han triat. Dins de l'aplicació es poden configurar les notificacions.
- Els professors podran, a part de visualitzar, modificar la descripció dels esdeveniments corresponents a l'assignatura que ells imparteixen. La idea és que qualsevol informació que vulgui fer arribar als alumnes ho pugui fer automàticament a través del esdeveniment de la propera classe i l'alumne rebi una notificació al mòbil indicant que hi ha hagut un canvi en un esdeveniment proper.
- El PAS s'encarrega de gestionar els calendaris. Queda per decidir si una modificació d'horaris d'una classe ho han de fer ells obligatòriament o també ho pot fer el professor.

Una correcta configuració d'aquests calendaris pot, amb relativament poc esforç, crear una via de comunicació Professor Alumne molt directa i ràpida i per part del PAS proporcionar la informació que es consideri important directa al mòbil de l'usuari.

10.2. Correu electrònic de l'EPS.

La creació d'una secció de correu en l'aplicació amb la conta de correu de la UdL seria també de força utilitat. Això podria facilitar la sincronització dels calendaris en el punt 10.1.

Si l'aplicació disposés d'un identificador únic com el correu es podria, en el moment que l'usuari la instal·la i configura el correu, adaptar automàticament a les assignatures en que està matriculat o imparteix.

10.3. Sincronització amb el Campus Virtual.

En l'enquesta es va rebre varies propostes de sincronitzar l'aplicació amb el Campus Virtual de la UdL. Si es pugués implementar de forma correcta potser seria la opció ideal ja que sumariem al Campus Virtual la immediatesa de comunicació que ara possiblement no té en lloc de crear un canal de comunicació paral·lel.

10.4. Implementació en multi-plataforma.

L'aplicació s'ha creat únicament per plataforma Android. Com a entitat pública i més tractant-se d'una universitat seria molt adequat implementar-la per a les altres plataformes més utilitzades actualment.

10.5. Elements d'accessibilitat pendents.

En l'apartat de Conclusions s'indica que hi ha hagut uns punts dels objectius d'accessibilitat que no s'han pogut assolir. En cas que en un futur projecte es modifiqui l'aplicació de l'EPS seria convenient tenir en compte aquests punts.

- Reconeixement de veu i sistema de lectura de pantalla. Implementar la possibilitat que sigui accessible l'aplicació per a invidents per poder interactuar plenament i extreure'n la informació desitjada.
- Canvi d'idioma. En la secció de preferències s'ha implementat el menú per realitzar aquest canvi però no s'ha creat els documents XML d'altres idiomes ni s'ha introduït el codi corresponent en les classes per poder assumir el canvi.
- Canvi de mida de lletra. Igual que l'idioma, la mida de lletra està configurat en la secció de preferències però no s'ha programat per a que sigui operatiu.

10.6. Llançament de l'aplicació

La part de llançament de l'aplicació no s'ha pogut realitzar en el termini de presentació del projecte. S'intentarà realitzar-ho posteriorment però, de totes maneres, s'inclou com a treball futur ja que és un punt pendent de realitzar.

Si es pot és recomanable fer la distribució a través de la plataforma de venda de Google "Play Store". En cas que això no sigui possible econòmicament o per qualsevol altre motiu s'haurà de buscar un canal de distribució alternatiu. Possiblement es pugui incloure un punt de descarrega a la pàgina web de l'EPS com a solució alternativa però s'haurà d'estudiar les possibles limitacions que pugui provocar la instal·lació d'una aplicació no certificada en els dispositius.

11. Bibliografía.

- [1] RIBAS LEQUERICA, Joan. *Desarrollo de aplicaciones para Android*. Madrid: Ediciones Anaya Multimedia (Grupo Anaya, S.A.), 2011. Colección Manuales imprescindibles.
- [2] BENBOURAHILA, Nazim. *Android 4, Principios del desarrollo de aplicaciones Java*. Traductor Rodríguez, Oncins. Barcelona: Ediciones ENI, Julio 2013. Colección Recursos Informáticos.
- [3] NEIL, Theresa. *Mobile Design Pattern Gallery*. USA: O'Reilly Books, March 2012.

12. Webgrafia

- [4] *Android Developers* [en línia]: <http://developer.android.com/index.html>
- [5] CUELLO, Javier “Simón” i VITTONI, José. *Diseñando apps para móviles*: [en línia] <http://www.appdesignbook.com/es/contenidos/> [consulta: 21/07/14].
- [6] ALBES, Luis. *El baúl de Android* [en línia]: <http://elbauldeandroid.blogspot.com.es/2014/01/settings.html> [consulta: 24/07/14].
- [7] Q&A sites. *StackOverflow* [en línia]: <http://stackoverflow.com/>
- [8] *VidasConcurrentes* [en línia]: <http://blog.vidasconcurrentes.com/android/tratamiento-de-xml-en-android-lectura-y-escritura/> [consultat: 16/07/14].
- [9] Hermanos Carrero. *Programación en castellano* [en línia]: http://programacion.net/articulo/tutorjava_nivel_basico_97/1 [consulta: 30/06/14]
- [10] GÓMEZ, Salvador. *Sgoliver.net blog* [en línia]: http://www.sgoliver.net/blog/?page_id=3011
- [11] NIELSEN, Jakob. *Website Response Times. Nielsen Norman Group*. [en línia] : <http://www.nngroup.com/articles/website-response-times/> [Publicat: 21/06/10]
- [12] IDC, *Analyze the Future, Smartphone OS Market Share Q2, 2014* [en línia]: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp> [consultat: 05/09/14]
- [13] GRIHO, Grup de Recerca en Interacció Persona Ordinador i Integració de Dades. *MPIu+a, Modelo de Proceso de la Ingeniería de la usabilidad y de la accesibilidad* [en línia]: <http://www.grihotools.udl.cat/mpiu+a/>

13. Annex I. Formulari de l'enquesta.

Dades personals.

Pregunta 1:

A quin col·lectiu pertany?

- Alumne de Grau
- Alumne de Màster
- Alumne de pla antic.
- Professor

Pregunta 2:

Quina titulació estàs cursant?

(Pregunta 1: Alumne de Grau)

- Grau en Enginyeria de l'Edificació
- Grau en Enginyeria Informàtica
- Grau en Enginyeria Mecànica
- Grau en Enginyeria Electrònica Industrial i Automàtica

(Pregunta 1: Alumne de Màster)

- Màster en Enginyeria de Programari Lliure
- Màster en Interacció Persona Ordinador
- Màster en Enginyeria Industrial
- Màster universitari en Ciències Aplicades a l'Enginyeria
- Màster en Enginyeria Informàtica

(Pregunta 1: Alumne de pla antic)

- Enginyeria Tècnica Informàtica Gestió
- Enginyeria Tècnica Informàtica Sistemes
- Enginyeria Tècnica Industrial Mecànica
- Arquitectura Tècnica
- Enginyeria Informàtica (2n cicle)

Dispositiu

Pregunta 3:

Disposes de mòbil intel·ligent? (telèfon mòbil amb pantalla tàctil i/o teclat que incorpora un navegador d'Internet i aplicacions que es poden descarregar) Quin Sistema Operatiu te?

- Blackberry OS
- iOS (iPhone)
- Android
- Symbian OS
- Windows
- Altres
- No disposo de mòbil intel·ligent

Hàbits d'ús

Pregunta 4:

Per a que utilitzeu el vostre dispositiu mòbil i amb quina freqüència ho feu?

(Valoració: Molts cops al dia, Diàriament, Sovint, Poc, Mai)

- Consum de continguts multimèdia (música, vídeos...)
- Accés a xarxes socials
- Agenda
- Cerca d'informació a Internet
- Consulta de correu electrònic

- Consultar mapes o altres aplicacions de geolocalització
- Consulta de pàgines web
- Descàrrega d'aplicacions
- Fer fotos
- Jugar
- Missatgeria instantània

Pregunta 5:

Quins apartats de la web de l'Escola politècnica Superior visiteu?

(Marcar les que visiteu habitualment)

- Notícies
- Agenda
- L'escola. Informació General
- L'escola. Organització
- L'escola. Instal·lacions
- L'escola. Departaments
- Informació acadèmica. Normatives
- Informació acadèmica. Plans d'estudi
- Informació acadèmica. Guies docents
- Informació acadèmica. Horaris i calendaris
- Informació acadèmica. Practiques
- Tràmits Secretaria. Matrícula
- Tràmits Secretaria. Altres tràmits de matrícula
- Tràmits Secretaria. Avaluació curricular
- Tràmits Secretaria. Altres tràmits administratius
- Apartat d'Estudis

Pregunta 6:

Quins d'aquests apartats creus que haurien de ser accessibles des d'un dispositiu mòbil? Teniu en compte que no tots els apartats poden requerir la

immediatesa de que es disposa al accedir des del mòbil o que, pel mateix motiu, estiguin sotmesos a diferents limitacions ja sigui per la mida de la pantalla o l'absència de teclat i/o ratolí.

(Valoració de 1 a 5 de forma que 1 equival a poc útil i 5 molt útil)

- Notícies
- Agenda
- L'escola. Informació General
- L'escola. Organització
- L'escola. Instal·lacions
- L'escola. Departaments
- Informació acadèmica. Normatives
- Informació acadèmica. Plans d'estudi
- Informació acadèmica. Guies docents
- Informació acadèmica. Horaris i calendaris
- Informació acadèmica. Practiques
- Tràmits Secretaria. Matrícula
- Tràmits Secretaria. Altres tràmits de matrícula
- Tràmits Secretaria. Avaluació curricular
- Tràmits Secretaria. Altres tràmits administratius
- Estudis (Informació sobre les diferents titulacions)

Pregunta 7:

Valora les següents eines segons creguis que serien útils (molt útil 5, poc útil 1) per accedir des del mòbil o no.

- Integració en el calendari del dispositiu els horaris de classe de les assignatures en que esteu matriculats de forma que al configurar l'aplicació al calendari del dispositiu us surti totes les assignatures.
- Comunicació en temps real de possibles incidències i canvis en les classes del dia.

- Indicació del temari a tractar en la sessió de classe següent de l'assignatura indicada.
- Personalització de l'aplicació de forma que se li indiqui per part de l'usuari sobre quines assignatures vol estar informat i quines no de forma que sol es rebrà avisos de les assignatures que es desitgi.

Pregunta 8:

Creus que l'aplicació hauria de tenir algun altre apartat que no hi sigui en els anteriors? Descriu qualsevol eina que creguis que sigui útil en l'aplicació ponderant-la segons la puntuació anterior.

(Quadre de text)

Pregunta 9:

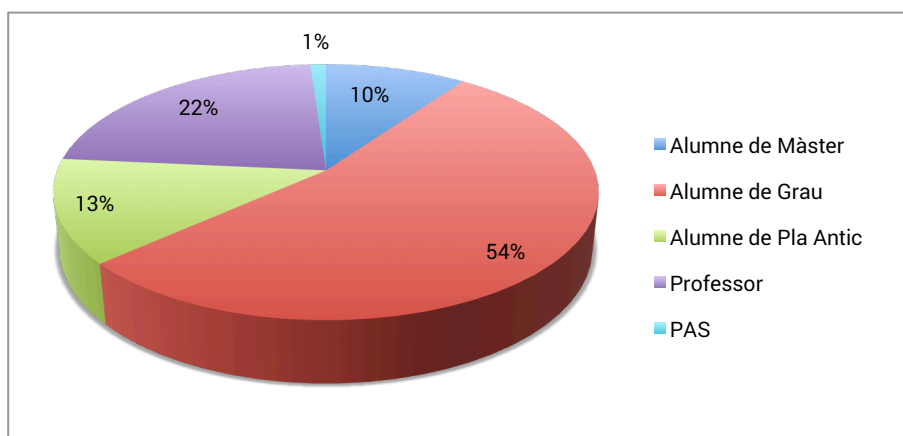
Propostes.

(Quadre de text)

14. Annex II. Resultats de l'enquesta.

1 - A quin col·lectiu pertanyes?

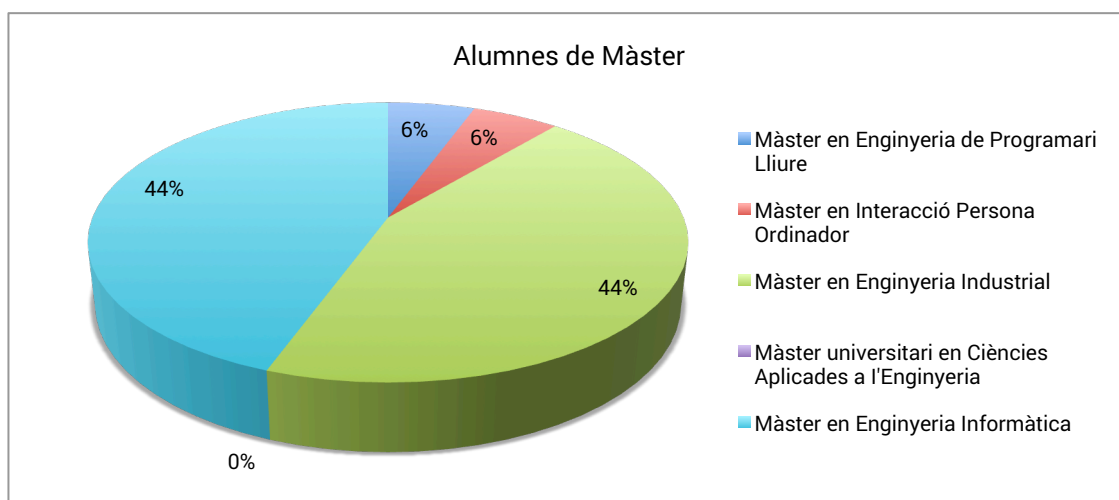
Respostes possibles	Quantitat
Alumne de Màster	18
Alumne de Grau	98
Alumne de Pla Antic	24
Professor	41
PAS	2



2 - Quina titulació està cursant?

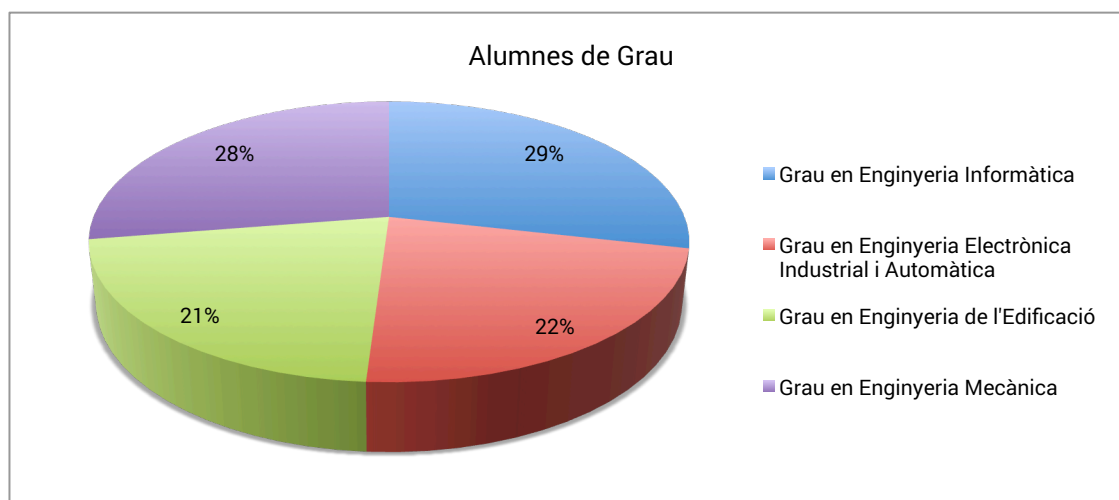
Alumnes de Màster.

Alumnes de Màster	Quantitat
Màster en Enginyeria de Programari Lliure	1
Màster en Interacció Persona Ordinador	1
Màster en Enginyeria Industrial	8
Màster universitari en Ciències Aplicades a l'Enginyeria	0
Màster en Enginyeria Informàtica	8



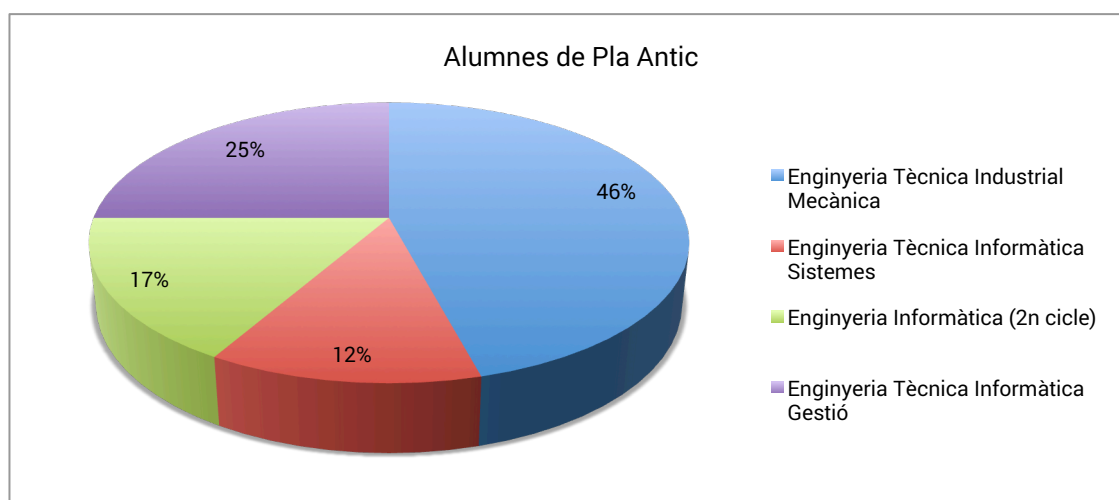
Alumnes de Grau.

Alumnes de Grau	Quantitat
Grau en Enginyeria Informàtica	28
Grau en Enginyeria Electrònica Industrial i Automàtica	22
Grau en Enginyeria de l'Edificació	21
Grau en Enginyeria Mecànica	27



Alumnes de Pla Antic.

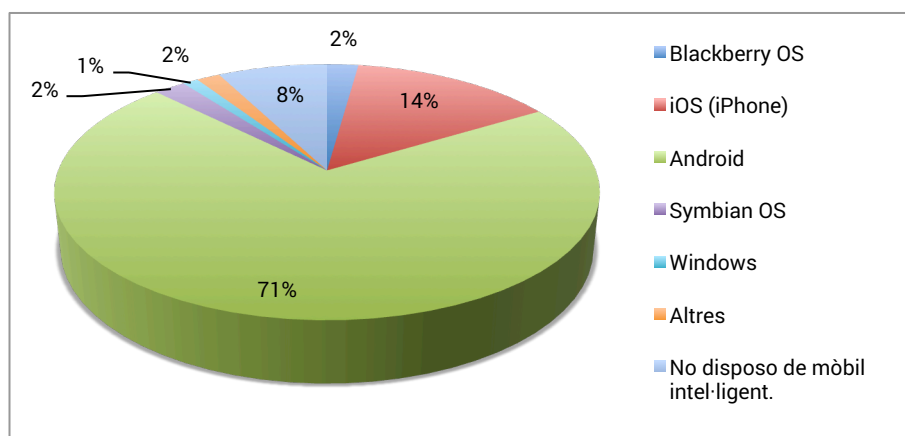
Alumnes de pla antic	Quantitat
Enginyeria Tècnica Industrial Mecànica	11
Enginyeria Tècnica Informàtica Sistemes	3
Enginyeria Informàtica (2n cicle)	4
Enginyeria Tècnica Informàtica Gestió	6



3 - Disposes de mòbil intel·ligent? (telèfon mòbil amb pantalla tàctil i/o teclat que incorpora un navegador d'Internet i aplicacions que es poden descarregar)

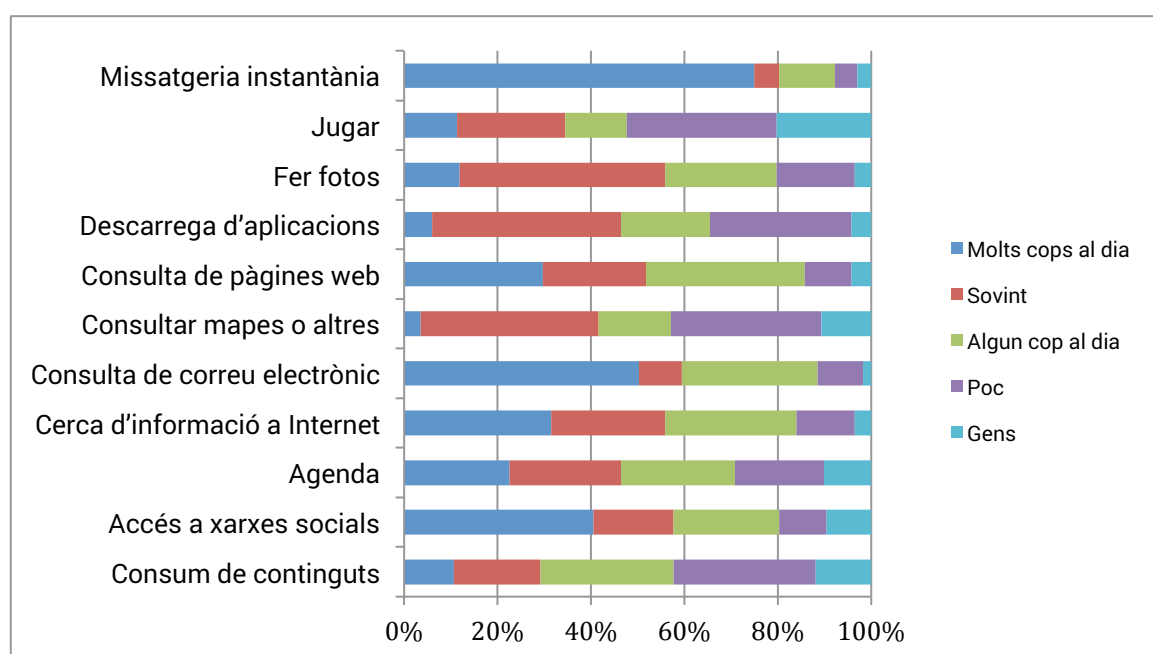
Quin Sistema Operatiu te?

Sistema Operatiu	Quantitat
Blackberry OS	4
iOS (iPhone)	26
Android	130
Symbian OS	4
Windows	2
Altres	3
No disposo de mòbil intel·ligent.	14



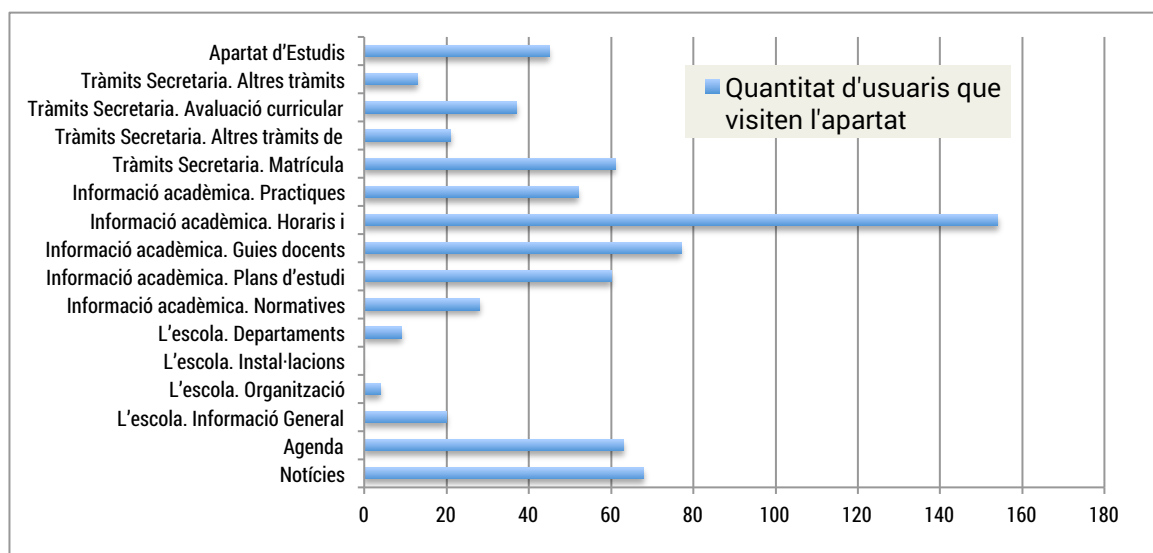
4 - Per a que utilitzeu el vostre dispositiu mòbil i amb quina freqüència ho feu?

Apartats	Molts cops al dia	Sovint	Algun cop al dia	Poc	Gens
Consum de continguts multimèdia (música, vídeos...)	18	31	48	51	20
Accés a xarxes socials	68	29	38	17	16
Agenda	38	40	41	32	17
Cerca d'informació a Internet	53	41	47	21	6
Consulta de correu electrònic	83	15	48	16	3
Consultar mapes o altres aplicacions de geolocalització	6	64	26	54	18
Consulta de pàgines web	50	37	57	17	7
Descàrrega d'aplicacions	10	68	32	51	7
Fer fotos	20	74	40	28	6
Jugar	19	39	22	54	34
Missatgeria instantània	126	9	20	8	5



5 - Quins apartats de la web de l'Escola politècnica Superior visiteu?

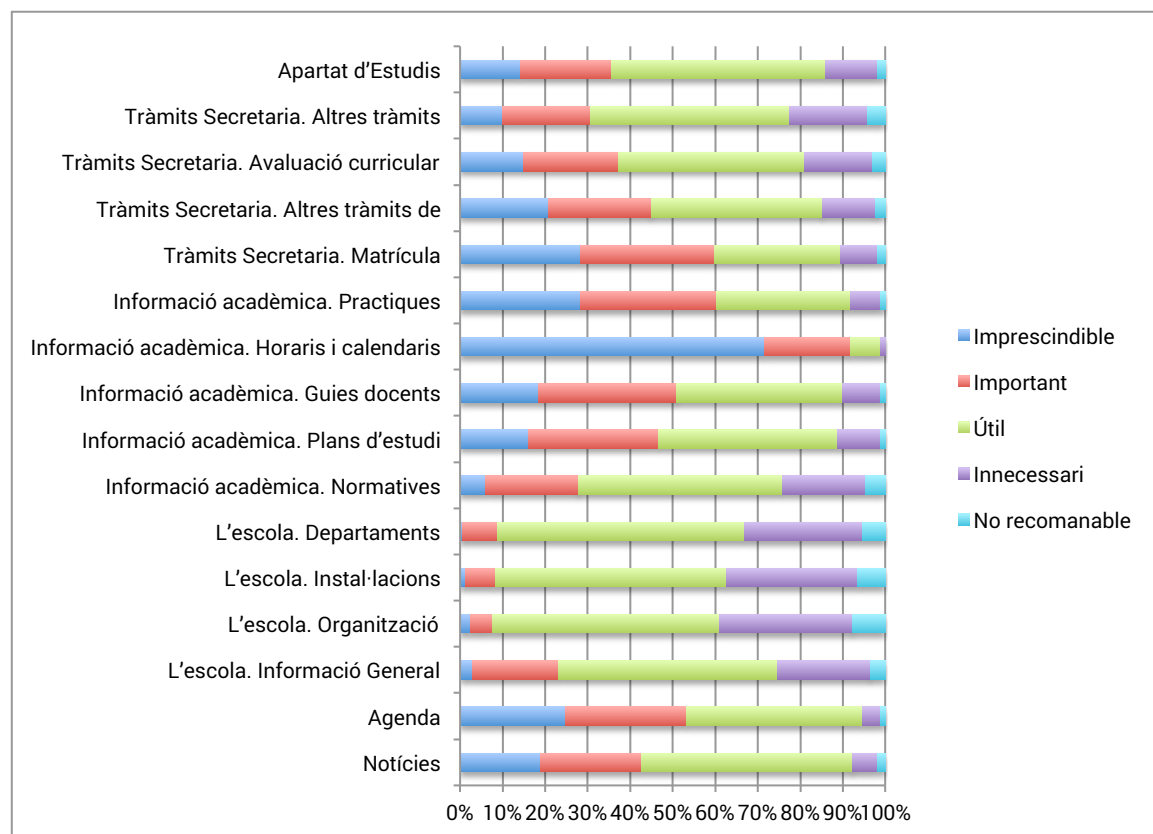
Apartat	Quantitat d'usuaris que visiten l'apartat
Notícies	68
Agenda	63
L'escola. Informació General	20
L'escola. Organització	4
L'escola. Instal·lacions	0
L'escola. Departaments	9
Informació acadèmica. Normatives	28
Informació acadèmica. Plans d'estudi	60
Informació acadèmica. Guies docents	77
Informació acadèmica. Horaris i calendaris	154
Informació acadèmica. Practiques	52
Tràmits Secretaria. Matrícula	61
Tràmits Secretaria. Altres tràmits de matrícula	21
Tràmits Secretaria. Avaluació curricular	37
Tràmits Secretaria. Altres tràmits administratius	13
Apartat d'Estudis	45



6 - Dels apartats de la pregunta anterior, quins creus que seria mes important poder accedir des d'un dispositiu mòbil?

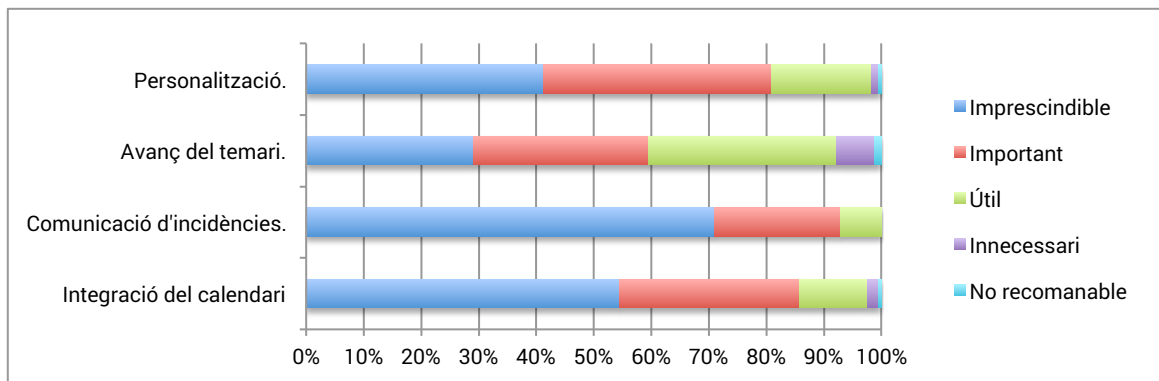
Apartat	Imprescindible	Important	Útil	Innecessari	No recomanable
Notícies	32	40	84	10	3
Agenda	42	48	70	7	2
L'escola. Informació General	5	34	87	37	6
L'escola. Organització	4	9	90	53	13
L'escola. Instal·lacions	2	12	92	52	11
L'escola. Departaments	1	14	98	47	9
Informació acadèmica. Normatives	10	37	81	33	8
Informació acadèmica. Plans d'estudi	27	52	71	17	2
Informació acadèmica. Guies docents	31	55	66	15	2

Informació acadèmica. Horaris i calendaris	121	34	12	2	0
Informació acadèmica. Practiques	48	54	53	12	2
Tràmits Secretaria. Matrícula	48	53	50	15	3
Tràmits Secretaria. Altres tràmits de matrícula	35	41	68	21	4
Tràmits Secretaria. Avaluació curricular	25	38	74	27	5
Tràmits Secretaria. Altres tràmits administratius	17	35	79	31	7
Apartat d'Estudis	24	36	85	21	3



7 - Valora les següents eines segons creguis que serien útils (molt útil 5, poc útil 1) per accedir des del mòbil o no.

Apartat	Imprescindible	Important	Útil	Innecessari	No recomanable
Integració en el calendari del dispositiu els horaris de classe de les assignatures en que esteu matriculats de forma que al configurar l'aplicació al calendari del dispositiu us surti totes les assignatures.	92	53	20	3	1
Comunicació en temps real de possibles incidències i canvis en les classes del dia.	120	37	12	0	0
Indicació del temari a tractar en la sessió de classe següent de l'assignatura indicada.	49	51	55	11	2
Personalització de l'aplicació de forma que se li indiqui per part de l'usuari sobre quines assignatures vol estar informat i quines no de forma que sol es rebrà avisos de les assignatures que es vulgui.	69	66	29	2	1



8 - Creus que l'aplicació hauria de tenir algun altre apartat que no hi sigui en els anteriors? Descriu qualsevol eina que creguis que sigui útil en l'aplicació ponderant-la segons la puntuació anterior.

- Un xat amb tots els usuaris d'una assignatura(excepte professors, per poder debatre sense por a represàlies)
- El apartat de recursos seria molt interessant, ja que així sabríem les ultimes actualitzacions per poder seguir el temari del professor/a.
- En la integració del calendari en el dispositiu, trobo que també seria important posar el dels exàmens, depenent de la titulació que es cursi.
- No
- Ponderaria "Important" una eina similar al "Directori" del Campus Virtual, tot i que probablement ho seria més d'útil encara si estigués actualitzada amb els despatxos dels professors i els seus horaris d'estar al despatx.
- no, crec que està tot lo mitjanament útil i necessari.
- No.
- Notificació de notes d'exàmens i assignatures a temps real.
- Integración dentro de la aplicación del Mail y apartados referentes a la información de campus virtual como : foros, presentación de trabajos, etc.
- Algun apartat on els professors de cada assignatura puguin posar les notes a mesura que es van realitzant activitats i exàmens, i que surti a temps real en el dispositiu.
- avis d'exàmens i entregues de practiques
- no

- calendario de exámenes
- Comunicació amb xat amb professor i alumnes classe
- Campus virtual
- Informació del períodes d'inici i finalització de tràmits administratius com pot ser matrícula, convalidacions, convocatòries de beques, modificacions, resolucions de convocatòries.
- No crec.
- Un possible apartat amb el correu dels professors de les assignatures a les quals un alumne està matriculat, d'aquesta manera seria molt més fàcil accedir-hi per poder enviar-li un missatge al professor
- Penso que si es compleixen els apartats anteriors serà un gran èxit
- Potser estaria bé algun espai on es poguessin deixar comentaris sobre els diferents graus i professors per tal d'informar sobre opinions que poden ser útils a alumnes futurs. També estaria bé algun espai on la gent que actualment està treballant en alguna cosa relacionada amb el que ha estudiat expliqués de que li ha servit estudiar la carrera que ha fet, i les possibles matèries que s'haurien de cursar en un futur segons la seva opinió per tal de que la universitat estigui més relacionada en l'ambient laboral del moment ja sigui en forma de realització de petits cursets o variant el pla d'estudi. A part també es podria informar sobre cursets i jornades que es facin a la UdL relacionades amb els estudis que realitzi l'usuari.
- Horarios de disponibilidad de cada profesor.
- L'expedient i notes de parcials
- Afegiria el calendari acadèmic, és a dir, no només els de les assignatures, sinó també el de l'EPS i la UdL.

- consulta del correu
- "Espacio para comunicación entre estudiantes y organización de eventos. Espacio para relación empresa-universidad. Bolsa de prácticas-trabajo.
- Incidències generals de la UdL i de la EPS (limitada a uns continguts determinats per evitar un excés d'informació).
- Hauria de poder rebre els missatges que fan els professor, per tal d'avisar als alumnes de les incidències durant les classes.
- Accés directe al correu d'alumne i del campus.

9 – Propostes

- El apartar de recursos seria molt interessant, ja que així sabríem les ultimes actualitzacions per poder seguir el temari del professor/a.
- Molts avisos relacionats amb canvis d'horari i altres incidències ja ens venen al correu dels alumnes, però donat que la configuració d'avisos dóna problemes en alguns dispositius, la possibilitat que aquesta aplicació incorpori avisos tant de l'escola com fins i tot de part dels professors, seria molt ben rebuda.
- Crec que el apartat mes important és el dels horaris, i els canvis de horari de ultima hora.
- Les notícies, incidències i altres és important que estiguin actualitzades al dia, ja que la web actual manca en aquest sentit
- Trobo que aquesta és una proposta molt interessant. Tan de bo es pugui dur a terme tal com la planeges. Ànims!
- Seria molt interesant poder configurar el correu de la UdL com configurem el correu de hotmail o gmail.

- bona idea, a veure si realment funciona
- Rapida De Carregar i K Pesi Poc
- Informació de cursos de formació no reglada, crèdits ECTS, i activitats de formació d'idiomes.
- Espero que al igual que la página oficial, también esté en Español, ya que muchos somos de fuera, ya que ambas lenguas han de ser vehiculares, y ya que también es la lengua oficial de Cataluña. Un saludo.
- Que no se lo curren mucho en la estética de la aplicación, lo que importara será el contenido.
- "Bon dia Miquel,

He marcat que sóc professor per poder fer l'enquesta ja que crec que és un FC força interessant, però no sóc professor ... sóc PAS de l'EPS i no ens has ficat enlloc. Ja se que no som un col·lectiu gaire nombrós (actualment 15 persones) ni gaire important, però estaria bé que almenys apareguéssim en la enquesta. Moltes gràcies"

- Fer-ho multi-plataforma, en la mesura del possible.
- "Comentari:
- La ponderació que fas, potser no t'aportarà la informació que necessitaràs en quantificar les respostes, i no ho interpretaràs adequadament. Quan les respostes són Molts cops al dia / Algun cop al dia / Sovint / Poc / Gens, no queda clara la diferència entre Sovint i Molts cops al dia, o entre Algun cop al dia i Sovint i Poc. Potser hauries de quantificar les vegades al dia que se'n fa us d'aquestes eines en cada cas. El que si que està clar és l'opció: ""Gens"".
- Si necessites més ajuda no dubtis en contactar.

15. Annex III. Prototip digital.



Figura 65. Menú

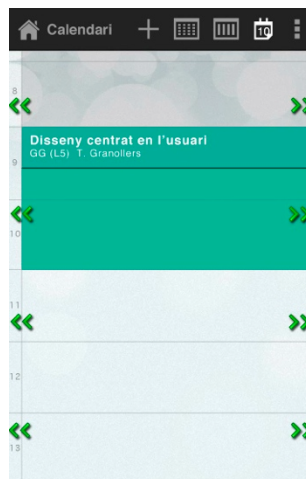


Figura 66. Calendari, vista dia.

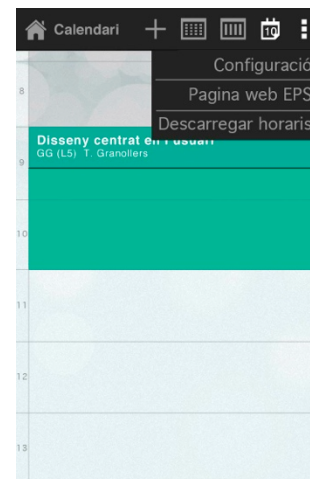


Figura 64. Menú de l'ActionBar desplegat en la interfície Calendari.



Figura 69. Calendari, vista mes.

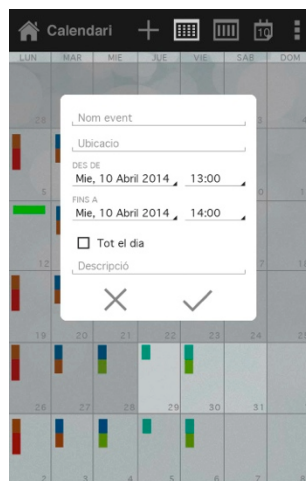


Figura 68. Calendari, nou esdeveniment.



Figura 67. Calendari, vista setmana.



Figura 72. Calendari, consulta d'un esdeveniment.



Figura 71. Secció de descarregues del calendari.

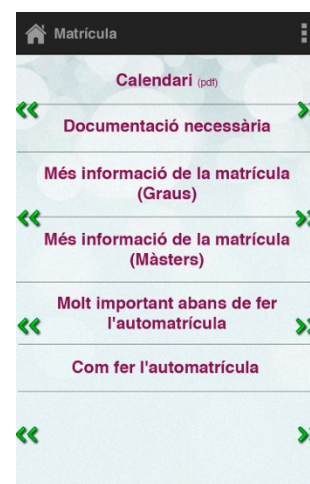


Figura 70. Matrícula.

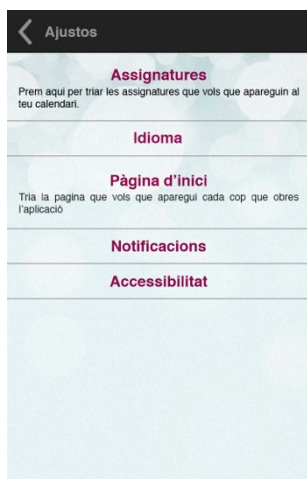


Figura 75. Secció d'Ajustaments.

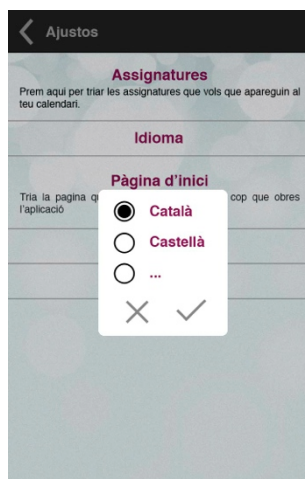


Figura 74. Ajustaments, selecció d'idioma.

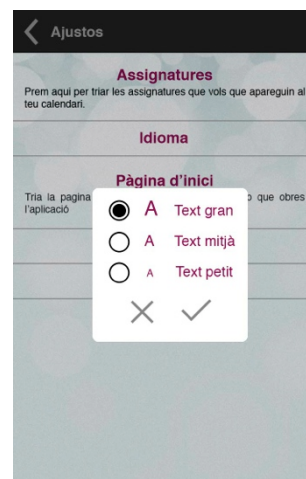


Figura 73. Ajustaments, selecció de la mida de text.

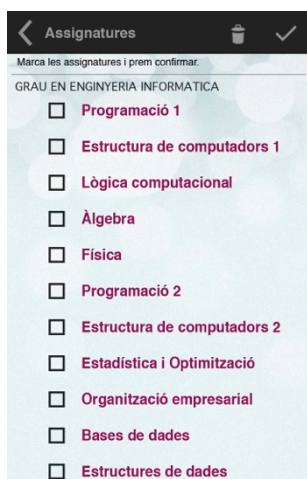


Figura 78. Ajustaments, selecció d'assignatures.

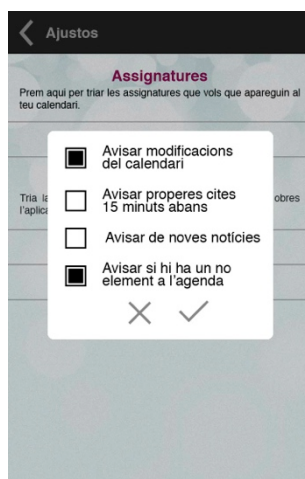


Figura 77. Ajustaments, configuració de les notificacions.



Figura 76. Guies Docents.



Figura 80. Notícies.



Figura 79. Agenda.